SIAM J. DISCRETE MATH.
Vol. 19, No. 1, pp. 1–18

AN IMPROVED ANALYSIS FOR APPROXIMATING THE SMALLEST k-EDGE CONNECTED SPANNING SUBGRAPH OF A MULTIGRAPH[*]

HAROLD N. GABOW[†]

Abstract. Khuller and Raghavachari [*J. Algorithms*, 21 (1996), pp. 434–450] present an approximation algorithm (the *KR algorithm*) for finding the smallest k-edge connected spanning subgraph (k-*ECSS*) of an undirected multigraph. They prove the KR algorithm has an approximation ratio < 1.85. We improve this bound to $\leq 1 + \sqrt{1/e} < 1.61$ (for odd k we modify the base case of the KR algorithm). This is the best-known performance bound for a combinatorial approximation algorithm for the smallest k-ECSS problem for arbitrary k. Our analysis also gives the best-known combinatorial performance bound for any fixed value of $k \geq 3$, e.g., for even k the approximation ratio is $\leq 1 + (1 - \frac{1}{k})^{k/2}$. Our analysis is based on a laminar family of sets (similar to families used in related contexts) which gives a better accounting of edges added in previous iterations of the algorithm. We also present a polynomial time implementation of the KR algorithm on multigraphs, running in the time for $O(nm)$ maximum flow computations, where n (m) is the number of vertices (edges, not counting parallel copies), respectively. This complements the implementation of Khuller and Raghavachari [*J. Algorithms*, 21 (1996), pp. 434–450] which uses time $O((kn)^2)$ and is efficient for small k.

Key words. approximation algorithms, network design, multigraphs, graph connectivity, edge connectivity, laminar family

AMS subject classifications. 05C40, 05C85, 68R10, 68W25, 68W40, 90B18, 90C27

DOI. 10.1137/S0895480102414910

1. Introduction.

Given a k-edge connected graph G, we seek a spanning subgraph with the fewest possible number of edges that is still k-edge connected. This is a natural problem in network design, e.g., the case $k = 1$ asks for a spanning tree. The problem is MAX SNP-hard for any fixed $k \geq 2$ [4, 7]. A number of approximation algorithms have been proposed. This paper achieves the best-known approximation guarantee for a combinatorial algorithm when G is an undirected multigraph. We do this by improving the analysis of the algorithm of Khuller and Raghavachari [14]. For the rest of the paper we refer to Khuller and Raghavachari's algorithm as the *KR algorithm*. *k-ECSS* stands for k-edge connected spanning subgraph. n and m denote the number of vertices and edges of the given graph, respectively.

Previous work. The problem of finding a smallest 2-ECSS has been widely investigated [15, 2, 17]. The best-known approximation ratio is 5/4, achieved by Jothi, Raghavachari, and Varadarajan [11] building on the approach of Vempala and Vetta [22]. For 2-ECSS parallel edges pose no problem and can essentially be ignored. However this is not the case for any higher connectivity $k \geq 3$. Gabow [6] gives a 3/2 approximation for the smallest 3-ECSS of a multigraph. Most of these 2- and 3-ECSS algorithms are based on depth-first search.

[*] Received by the editors September 19, 2002; accepted for publication (in revised form) September 7, 2004; published electronically June 3, 2005. A preliminary version of this paper appeared as *Better performance bounds for finding the smallest k-edge connected spanning subgraph of a multigraph*, Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2003, pp. 460–469.

http://www.siam.org/journals/sidma/19-1/41491.html

[†] Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80309-0430, (hal@cs.colorado.edu).

Cheriyan and Thurimella [3] present an elegant $1 + 2/(k+1)$ approximation algorithm for the smallest k-ECSS that is valid for arbitrary k under the assumption that the given graph is simple. Their approach is based on an analogue of a theorem of Mader on k-node connected graphs. This analogue holds for simple graphs. In section A.1 of the appendix, we show that on multigraphs the algorithm of [3] has performance ratio exactly 2. (More precisely, 2 is an upper bound on the approximation ratio, and for any fixed $k \geq 2$ there are arbitrarily large graphs where the ratio is arbitrarily close to 2.)

Several algorithms based on linear programming have been presented. Karger [12] uses randomized rounding to get a smallest k-ECSS algorithm for multigraphs with performance ratio $1 + O(\sqrt{(\log n)/k})$. This bound is of interest when $k >> \log n$. Goemans, Tardos, and Williamson [10] proposed rounding up an extreme point. Recently this was shown to achieve performance ratio $1 + 2/k$ [8]. This performance bound is more accurate than the bound achieved in this paper. However, the approach requires solving a large linear program, leading to a high running time. Therefore, fast combinatorial algorithms for k-ECSS are still attractive [21, p. 102].

The simplest known way to approximate the smallest k-ECSS is by a collection of k maximal spanning forests. This algorithm achieves performance ratio 2 on any multigraph. Nagamochi and Ibaraki [18] show how to find the desired forests in time $O(n(m + n \log n))$ on multigraphs.

Khuller and Raghavachari [14] gave the first algorithm for smallest k-ECSS that achieves approximation ratio better than 2. The KR algorithm is based on depth-first search. They prove the approximation ratio is strictly less than 1.85 on any multigraph. Their implementation uses time $O((kn)^2)$. Fernandes [4] improves the analysis of the KR algorithm on simple graphs, showing the approximation ratio is at most 1.75, and at most 1.7 for large enough k. The proofs of both bounds use the assumption of simplicity in crucial ways [4, see Facts 3.3 and 4.4].

Our contribution. This paper improves the analysis of the KR algorithm. We show the approximation ratio is at most $1 + \sqrt{1/e} < 1.61$ for arbitrary multigraphs. This is the best-known ratio for the smallest k-ECSS for any combinatorial algorithm. For every fixed connectivity $k > 1$ the ratio is better, specifically, $1 + (1 - \frac{1}{k})^{k/2}$ for k even and $1 + (1 - \frac{1}{k})^{(k-3)/2}(1 - \frac{3/2}{k})$ for k odd. These are the best-known ratios for combinatorial algorithms for every fixed $k \geq 3$.

Our approach uses a laminar family of sets introduced in [1, 5] and also used in Cheriyan and Thurimella's analysis [3]. This enables more accurate accounting than [14, 4]. Specifically, our estimate of the number of edges added in a given step of the KR algorithm takes every previously added edge into account.

To achieve the desired performance bound for odd k we use the recent 3-ECSS algorithm of [6]. The $1 + \sqrt{1/e}$ upper bound is proved using a simple bound from [6]. Our best bound for fixed odd k depends on extending the analysis of the 3-ECSS algorithm. Since this extension involves detailed knowledge of the algorithm it has been added as an appendix to [6].

Our upper bound $1 + \sqrt{1/e} < 1.61$ can be compared to a 1.5 lower bound: Section A.2 of the appendix gives an example adopted from [15] that shows for every even k the approximation ratio can be arbitrarily close to $3/2$. The appendix gives a similar example for odd $k > 1$, although for simplicity we only consider the original KR algorithm and not our modified version.

We also provide a polynomial-time implementation of the KR algorithm for weighted graphs, i.e., multigraphs where each edge has an integral value specifying its multiplicity. The running time is dominated by $O(nm)$ max flow computations, specifically time

$$O(\min\{(nm)^2 \log_b n,\ \mu nm^2 \log(n^2/m) \log k\}),$$

where $b = m/(n \log n) + 2$, $\mu = \min\{m^{1/2}, n^{2/3}\}$. The analysis uses a potential function to show the number of "distinct" iterations of the KR algorithm is polynomial regardless of k.

Organization of the paper. Section 2 is devoted to a recurrence that arises in our analysis. Section 3 reviews the KR algorithm and presents our analysis. Section 4 gives the polynomial-time implementation for multigraphs. The appendix gives the lower bound examples for the algorithm of Cheriyan and Thurimella and the KR algorithm. This section closes with terminology and notation.

Terminology. We often denote a singleton set $\{x\}$ by x. A family of sets is *laminar* if every two sets in the family are either disjoint or one contains the other.

All graphs are undirected. Multiple edges are always allowed in a graph (but not self-loops). We often denote an edge by juxtaposing its two vertices, as in vw. For multigraphs this notation need not designate a unique edge but this will not cause any confusion. We do not distinguish between a subset of edges and the spanning subgraph induced by those edges. For a simple graph G and an integer c, $c \cdot G$ denotes the multigraph constructed from G by giving every edge multiplicity c.

Take a (multi)graph $G = (V, E)$. Let X be a set of vertices. An edge with exactly one vertex in X *leaves* X. $d(X)$ denotes the number of edges leaving X. In this notation every edge is counted according to its multiplicity. For distinct vertices x, y, $\lambda(x, y)$ denotes the smallest number of edges whose removal disconnects x and y. If S is a set of edges, then appending S as the last argument to d or λ means the graph of interest is the spanning subgraph with edge set S, e.g., $d(X, S), \lambda(x, y; S)$.

Graph G is *k-edge connected* if $\lambda(x, y) \geq k$ for all $x \neq y$. For a given G and k with G k-edge connected, OPT is a smallest set of edges forming a k-ECSS of G.

Given a spanning tree, a nontree edge *covers* every edge in its fundamental cycle, while a tree edge covers itself. A *(maximal) dfs forest* consists of a depth-first spanning tree of each connected component. It is sometimes convenient to assume back edges are directed upwards, i.e., from the deeper end to the shallower end.

2. A recurrence.

For arbitrary real numbers α, a, b and for $c \neq 1$, consider the recurrence

$$(1) \qquad \begin{aligned} R_1 &= \alpha, \\ R_{i+1} &= a + bi + cR_i \quad \text{for all } i \geq 1. \end{aligned}$$

The solution is

$$R_i = c^{i-1}\alpha + a\frac{1 - c^{i-1}}{1 - c} + \frac{b}{1 - c}\left(i - \frac{1 - c^i}{1 - c}\right).$$

This solution can be derived using the identities for $j \geq 1$, $\sum_{i=0}^{j-1} c^i = \frac{1-c^j}{1-c}$, and $\sum_{i=0}^{j-1}(j - i)c^i = (j + 1 - \frac{1-c^{j+1}}{1-c})/(1 - c)$. Alternatively it can be verified by an easy induction.

We are interested in the case

$$(2) \qquad a = \frac{1}{k} + \frac{\epsilon}{k^2} \text{ for } \epsilon \in \{0, 2\}, \quad b = \frac{2}{k^2}, \quad c = 1 - \frac{1}{k}.$$

Using $1/(1-c) = k$ and some algebra we get for this case

$$R_i = c^{i-1}\left(1+\alpha-\frac{2+\epsilon}{k}\right)+\frac{2i+\epsilon}{k}-1. \tag{3}$$

Equation (4) below gives 3 particular values of R_i. (4a) is for k even and (4b)–(4c) are for k odd. We are still assuming (2), as well as $k > 1$. The values (4) all follow from (3) by simple algebra:

$$R_i = \begin{cases} (1-\frac{1}{k})^{k/2}, & i = k/2 \quad a = \alpha = 1/k, \quad \text{(4a)} \\ (1-\frac{1}{k})^{(k-1)/2} - \frac{1}{k}, & i = (k-1)/2 \quad a = \alpha = 1/k, \quad \text{(4b)} \\ (1-\frac{1}{k})^{(k-3)/2}\left(1+\alpha-\frac{4}{k}\right)+\frac{1}{k}, & i = (k-1)/2 \quad a = 1/k + 2/k^2. \quad \text{(4c)} \end{cases}$$
(4)

Now assume we have $\alpha = r/k$ for some constant $r \le 8/3$. This includes all the values we are interested in. Then all 3 quantities of (4) satisfy $\lim_{k\to\infty} R_i = \sqrt{1/e} \approx 0.606^+$. The quantity of (4a) increases with k. For (4b) and (4c) we will be interested in those quantities ignoring the last term $\pm 1/k$. The numerical values in the next section (Table 1) will show that for $k \ge 3$, the (4b) quantity (without $1/k$) decreases with k while the (4c) quantity (without $1/k$) increases with k. Our theorem requires a proof of this last relation for (4c). The proof is omitted since the numerical evidence is convincing and more meaningful. The interested reader can supply the argument, using logarithmic differentiation and Taylor series to show the computed derivative (w.r.t. k) is positive for $k \ge 3$ (and $r \le 8/3$).

3. Analysis of the KR algorithm. We begin this section by reviewing the KR algorithm. Section A.2 of the appendix illustrates the algorithm on an example. The input to the KR algorithm is a k-edge connected multigraph $G = (V, E)$. The output is a k-ECSS having a small number of edges.

We call the basic operation of the KR algorithm a *2-step*. The purpose of a 2-step is to increase the edge-connectivity of the current solution graph by 2. More precisely a 2-step starts with an h-ECSS H and adds a set of edges $F \cup B$ to get an $(h+2)$-ECSS $H \cup F \cup B$. F is a maximal dfs forest of $G - H$. It is easy to see that $F \cup H$ is $(h+1)$-edge connected. B is a set of back edges of the depth-first search. B is formed by traversing the forest F bottom-up and adding edges according to the following rule: When we go from a vertex v to its parent p, check if $\lambda(v, p; H \cup F \cup B) < h + 2$ (here B refers to the current set B). If so, add to B a back edge that goes from a descendant of v to a vertex closest to the root of the current tree.

It is proven in [14] that such a back edge always exists, and furthermore $H \cup F \cup B$ is $(h+2)$-edge connected at the end of the 2-step.

The overall KR algorithm starts with a spanning subgraph of no edges and performs $\lfloor k/2 \rfloor$ 2-steps. If k is odd, then an additional spanning forest is added at the end.

As just described, every 2-step in the KR algorithm starts with h even. However, the analysis of [14] remains valid if h is odd, and we shall use 2-steps that start with h odd. Also, it is important to bear in mind that in general F will not be a spanning tree but rather a forest. Certainly in the first iteration F is a spanning tree, but even in the second iteration F may consist of a large number of trees.

The analysis of [14] uses a property similar to the idea of "tree carvings" that was introduced in [15] for approximating the 2-ECSS problem. To state it, in any 2-step, let F^* be the set of edges vp of F that force a back edge e to be added to B (i.e., v and p are only $(h+1)$-edge connected when vp is traversed).

Lemma 1 (Khuller and Raghavachari). *In any 2-step, every edge of $G-H$ covers at most one edge of F^*.*

Proof. This follows from the fact that each edge of B is chosen to be directed to the shallowest possible vertex. For details see [14, proof of Lemma 3.6]. □

Our analysis, like previous ones, centers around estimating the number of back edges added by each 2-step. We use a laminar family of sets that covers every edge of F^*. A similar family is used in [3] but our family enjoys additional structural properties. For the following lemma fix an arbitrary 2-step. See Figure 1 for an example.

Lemma 2. *There is a laminar family of vertex sets $\mathcal{X} = \{X_f : f \in F^*\}$ such that*

(a) *each edge $f \in F^*$ leaves X_f and $d(X_f, H \cup F) = h+1$,*

(b) *each edge of $G-H$ leaves at most one set of $\mathcal{X}$,*

(c) *f is the unique edge of F leaving X_f; the edge of B that covers f leaves X_f and no other set of $\mathcal{X}$.*

Proof. We start by invoking a well-known fact discovered by Cai [1] and, independently in more general form, Frank [5]: Every k-edge connected (multi)graph has a laminar family $\mathcal{L}$ of vertex sets such that every set $X \in \mathcal{L}$ has $d(X) = k$ and every edge xy with $\lambda(x,y) = k$ leaves at least one set of $\mathcal{L}$. (This is proved by first applying an uncrossing argument and then converting the cross-free family to laminar.) Apply the fact to the $(h+1)$-edge connected graph $H \cup F$. Take any edge $f = xy \in F^*$. It is easy to check that $\lambda(x, y; H \cup F) = h+1$. Hence f leaves some set $X_f \in \mathcal{L}$. (If f leaves more than one set of $\mathcal{L}$ choose one of them arbitrarily to be X_f.) Define $\mathcal{X} = \{X_f : f \in F^*\}$. This laminar family satisfies (a).

Consider any edge $f \in F^*$. Since H is h-edge connected, it is easy to see that $d(X_f, H) = h$ and f is the only edge of F leaving X_f. (This gives the first part of (c).) Since F is a maximal spanning forest of $G-H$, an edge of $G-H$ leaves X_f if and only if it covers f. Now (b) follows from Lemma 1. The rest of (c) also follows. □

We introduce some more notation concerning a 2-step. An edge of H is *external* if it leaves some set of $\mathcal{X}$ and *internal* if it does not (see Figure 1). EXT (INT) denotes the set of external (internal) edges of H. Note that an external edge can leave an arbitrary number of sets of $\mathcal{X}$.

Lemma 3. *$|OPT| \ge (k-h)|B| + |EXT|$.*

Proof. Consider any set $X_f \in \mathcal{X}$. OPT contains $\ge k-h$ edges of $G-H$ that leave X_f (since Lemma 2(a) implies $d(X_f, H) = h$). Choose exactly $k-h$ such edges and add them to a set O. Doing this for every $f \in F^*$ makes O a subset of OPT with $|O| = (k-h)|B|$ and $d(X_f, O \cup H) = k$. This follows since an edge added for X_f does not leave any other set X_g (Lemma 2(b)). Furthermore $|F^*| = |B|$.

Initialize a set J to EXT. Observe that

$$d(X, O \cup J) = k \text{ for every } X \in \mathcal{X}. \tag{5}$$

We will now repeatedly remove 1 edge from J and add ≥ 1 new edge of OPT to O. Doing this for every edge of J will enlarge O to a subset of $\ge (k-h)|B| + |EXT|$ distinct edges of OPT. Clearly this implies the inequality of the lemma. In order to accomplish this we will maintain two invariants throughout the procedure, (5) and

$$OPT \subseteq O \cup J \cup \overline{EXT}. \tag{6}$$

Condition (6) obviously holds when J is initialized to EXT.

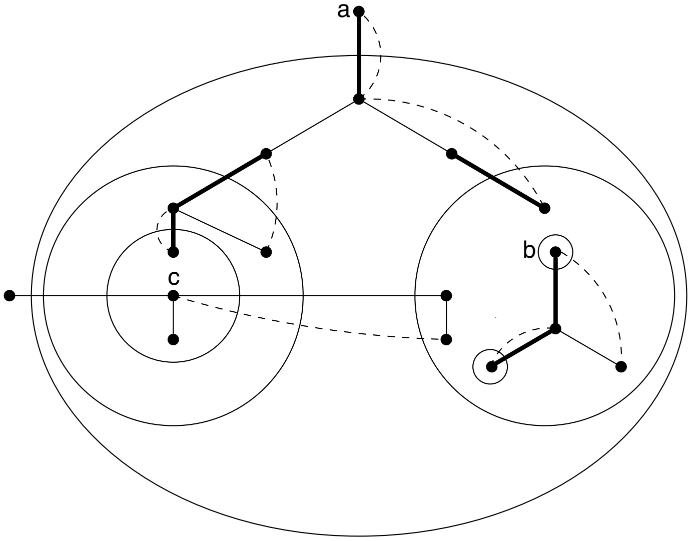


FIG. 1. *A laminar family $\mathcal{X}$ with 6 sets. Let T_x denote a tree rooted at x. The depth-first forest F contains trees T_a and T_b. F^* contains the 6 heavy edges and B contains the corresponding 6 dashed edges. H contains many edges, including another copy of T_a and T_b, and the tree T_c with a corresponding back edge. These 3 trees of H have $3, 1,$ and 2 internal edges, respectively. T_c has 2 external edges, and its back edge is external too.*

Consider an external edge e remaining in J. If $e \in OPT$ simply transfer e from J to O. This does not change $O \cup J$ so (5)–(6) continue to hold.

Now assume $e \notin OPT$. e leaves one or more sets $X \in \mathcal{X}$. For each such X choose one edge $d \in OPT - (O \cup J)$ that leaves X. Such a d exists by (5) and the fact that OPT is k-edge connected. Let D be the set of all chosen edges d. We will change O and J to $O' = O \cup D$ and $J' = J - e$, respectively.

Clearly $|O'| > |O|$ as promised. O' and J' satisfy (6) since $e \notin OPT$. We must check that $O' \cup J'$ satisfies (5). It is obvious that any $X \in \mathcal{X}$ has $d(X, O' \cup J') \geq k$ but we must check that equality holds, i.e., each $d \in D$ leaves exactly one set of $\mathcal{X}$. Equation (6) shows that $d \notin EXT$. Since d leaves a set of $\mathcal{X}$ it belongs to $G - H$. Now Lemma 2(b) shows that d leaves exactly one set of $\mathcal{X}$. □

The rest of this section looks at the 2-steps that built up the current h-ECSS H. If h is even let F_j and B_j, $j = 1, \ldots, h/2$, denote the sets of edges F and B that were added in the jth 2-step, respectively. So $H = \bigcup_{j=1}^{h/2} F_j \cup B_j$. As already mentioned, when k is odd we will discuss variants of the algorithm that make h odd. For these odd h we sometimes use the preceding definition of F_j and B_j. But in the main variant for k odd we make a small change in the definition of F_j and B_j (see the discussion after Lemma 7). Note also that, regardless of the parity of h, we will continue to refer to sets of the current iteration (e.g., F^*, B, INT) without affixing a subscript (by rights B would be referred to as $B_{h/2+1}$ etc.).

The notation $c(K)$ denotes the number of connected components of the graph K. By convention if K is a set of edges of G we consider K to be a spanning subgraph of G in the notation $c(K)$, and each isolated vertex contributes one to $c(K)$.

Observe that B_j is a forest. This follows from a simple fact about dfs trees.

PROPOSITION 4. *A set of back edges of a dfs tree is a forest if it contains ≤ 1 back edge directed from each vertex.*

Proof. We prove the contrapositive. Let x be a vertex of greatest depth in a cycle of back edges. Both cycle edges incident to x are back edges directed from x. □

In the statement of the next two results, j ranges over all possibilities $j = 1, \ldots, h/2$. Recall that our notational convention implies the sets F_j and B_j were constructed in an iteration prior to the current iteration, which constructs sets F and B. Note also that we will use a slightly stronger version of the next lemma when we treat odd values of k: Specifically, the lemma holds even if A is any acyclic set of back edges of F_j.

LEMMA 5. *For A an acyclic subset of $F_j \cup B_j$, $c(A \cap INT) \geq |B| + c(F_j)$.*

Example. In Figure 1 let F_j consist of copies of T_a, T_b, and T_c; as before, F consists of T_a and T_b. The right-hand side of the lemma is $|B| + c(F_j) = 6 + 3 = 9$. T_a and T_b contain 8 and 4 vertices, respectively. For $A = T_c$, $c(A \cap INT) = 8 + 4 + 3 = 15$. For $A = F_j$, $c(A \cap INT) = 5 + 3 + 3 = 11$.

Proof. Consider the set of edges $S = (A \cap INT) \cup F^*$. We first show S is a forest. By way of contradiction assume S contains a cycle C. Since A is acyclic, C contains an edge of F^*, say f. f is the unique edge of C leaving X_f (Lemma 2(c)). But this contradicts the fact that any cycle leaves X_f an even number of times.

Next we claim that every edge of S joins two vertices in the same tree of F_j. This holds by definition for A. For an edge $f = xy \in F^*$, x and y are in the same tree of F. Since F_j is chosen as a maximal spanning forest, x and y must also be in the same tree of F_j.

The claim implies that $c(S) \geq c(F_j)$. Thus $c(A \cap INT) \geq |F^*| + c(F_j)$, and the lemma follows. □

COROLLARY 6. (a) $|B_j \cap INT| + |B| + c(F_j) \leq n$. (b) $|F_j \cap EXT| \geq |B|$.

Proof. Recall that any acyclic set of edges R has $|R| + c(R) = n$.

(a) The initial observation and Lemma 5 give

$$n = |B_j \cap INT| + c(B_j \cap INT) \geq |B_j \cap INT| + |B| + c(F_j)$$

as desired.

(b) The initial observation and Lemma 5 give

$$|F_j| + c(F_j) = n = |F_j \cap INT| + c(F_j \cap INT) \geq |F_j \cap INT| + |B| + c(F_j).$$

(b) follows since F_j is partitioned into $F_j \cap EXT$ and $F_j \cap INT$. □

LEMMA 7. *If k is even the KR algorithm achieves approximation ratio $1 + (1 - \frac{1}{k})^{k/2}$.*

Proof. Let S_i be the total number of back edges added in the first i 2-steps (i.e., the steps that achieve $2i$ connectivity). We shall prove the recurrence

$$\begin{aligned} S_1 &\leq \tfrac{1}{k}|OPT|, \\ S_{i+1} &\leq \left(\tfrac{1}{k} + \tfrac{2i}{k^2}\right)|OPT| + \tfrac{k-1}{k}S_i \quad \text{for all } k/2 > i \geq 1. \end{aligned} \tag{7}$$

It is easy to see that (7) implies $S_i/|OPT|$ is upper-bounded by the quantity R_i of recurrence (1) with the values (2) and $\epsilon = 0$, $\alpha = 1/k$. So (4a) shows the total number of back edges added by the algorithm is $S_{k/2} \leq (1 - \frac{1}{k})^{k/2}|OPT|$. Besides these back edges the algorithm adds $k/2$ dfs forests. The degree lower bound $|OPT| \geq kn/2$ implies the forests have total size at most $|OPT|$. The lemma follows.

It remains to prove (7). The base case $S_1 = |B_1| \leq |OPT|/k$ is the carving lower bound [13, Theorem 6.2]. Alternatively, the base case follows by taking $H = \emptyset$ in Lemma 1 and concluding that OPT contains $\geq k|B_1|$ edges.

Table 1

Values of our approximation ratio for selected k.

k	2	4	6	8	10	20	100	1000	10000
ratio	1.5	1.562^+	1.578^+	1.586^+	1.590^+	1.598^+	1.605^+	1.6063^+	1.6065^+

k	3	5	7	9	11	21	101	1001	10001
KR	1.666^+	1.64	1.629^+	1.624^+	1.620^+	1.613^+	1.608^+	1.6066^+	1.6065^+
KRs	1.555^+	1.586^+	1.594^+	1.598^+	1.600^+	1.603^+	1.606^+	1.6064^+	1.6065^+
KR3	1.5	1.56	1.577^+	1.585^+	1.589^+	1.598^+	1.604^+	1.6063^+	1.6065^+

Next we prove the upper bound on S_{i+1}. We will apply our analysis of the 2-step that enlarges H from an h-ECSS to an $(h+2)$-ECSS. Take $h = 2i$. Just as before we have sets $B_1, \ldots, B_i$ and the current set B, as well as the current sets INT and EXT. By definition

$$S_i = \sum_{j=1}^{i} |B_j|, \tag{8a}$$

$$S_{i+1} = S_i + |B|. \tag{8b}$$

The degree lower bound shows $|OPT| \ge kn/2$. Combining this with Corollary 6(a) shows

$$|B_j \cap INT| + |B| + c(F_j) \le 2|OPT|/k, \quad j = 1, \ldots, i. \tag{9}$$

We will discard the term $c(F_j)$.

Lemma 3 shows $|OPT| \ge (k-2i)|B| + |EXT|$. To lower bound the right-hand side note that EXT is partitioned into sets $F_j \cap EXT$ and $B_j \cap EXT$, $j = 1, \ldots, i$. Using Corollary 6(b) to lower bound the sizes of the first group of sets gives

$$(k-2i)|B| + \sum_{j=1}^{i} (|B| + |B_j \cap EXT|) \le |OPT|. \tag{10}$$

Add the $i+1$ inequalities (9)–(10) and use the fact that B_j is partitioned into sets $B_j \cap INT$ and $B_j \cap EXT$ to get

$$k|B| + \sum_{j=1}^{i} |B_j| \le (1 + 2i/k)|OPT|.$$

The sum on the left-hand side equals S_i by (8a). Adding $(k-1)S_i$ to both sides and using (8b) gives

$$kS_{i+1} \le (1 + 2i/k)|OPT| + (k-1)S_i.$$

This is equivalent to the desired relation of (7). □

The bound of Lemma 7 increases monotonically with k and is always $< 1+\sqrt{1/e} = 1.60653^+$. Selected values are given in the first 2 rows of Table 1.

We turn to odd values of k. The original KR algorithm for odd k executes the same 2-steps as the algorithm for $k-1$, after which it adds a spanning forest to achieve connectivity k. The analysis of Lemma 7 shows the number of back edges S_i satisfies recurrence (7), so the total number of back edges is $|OPT|$ times (4b). There are $(k+1)/2$ dfs forests, contributing a total of $\leq (1+1/k)|OPT|$ edges. We conclude the approximation ratio is $1+(1-\frac{1}{k})^{(k-1)/2}$. This quantity is illustrated in the row labelled "KR" of Table 1. It approaches $1+\sqrt{1/e}$ from above.

To achieve an approximation ratio that is always below $1+\sqrt{1/e}$ we use the following version of the KR algorithm: Initialize the solution subgraph to a "good" 3-ECSS. Then execute $(k-3)/2$ 2-steps. It remains to choose the base 3-ECSS.

The simplest choice is to use the KR algorithm: Initialize H to a dfs tree of G. Then do a 2-step to enlarge H to a 3-ECSS. Our analysis of this variant leads to the same bound as the original KR algorithm. The analysis hinges on the fact that the first 2-step adds $\leq |OPT|/k$ back edges. The total number of back edges remains $|OPT|$ times (4b). Further details are left to the reader.

To improve the bound we use the algorithm of [6], which we refer to as the ear algorithm. This algorithm finds a 3-ECSS A of a multigraph, achieving approximation ratio 3/2. The algorithm works in three phases which we now describe.

Phase I initializes A to a dfs tree, which we denote as F for consistency with a 2-step. Phase II begins by making A 2-edge connected by adding a set of back edges B'. To do this it first uses the 2-step procedure to construct B and the forcing edges F^*. It chooses the set B' in a top-down pass that guarantees each edge of B' covers a distinct edge of F^*. The edges of B' are called *long ear* edges. To describe the rest of Phase II we recall that the algorithm maintains a partition of V into sets of vertices that are known to be 3-edge connected in A, called "t-sets." The second part of Phase II adds edges called *short ears*. Each short ear merges at least 3 t-sets into 1. Phase III makes A 3-edge connected by adding a spanning forest of the t-sets.

For our analysis we partition the final 3-ECSS A into the spanning tree F plus 2 forests: B_0 contains the long ear edges. It is acyclic, by Proposition 4 and Lemma 1. B_1 consists of the short ears plus the spanning forest of Phase III. It is acyclic, since each edge merged ≥ 2 distinct t-sets. Take a value α such that $|B_0|+|B_1| \leq \alpha|OPT|$.

Let *KR*3 denote the k-ECSS approximation algorithm with the ear algorithm used for initialization. For $i \geq 1$ let S_i be the total number of back edges added in the first i steps of the algorithm (i.e., the initialization step and the first $i-1$ 2-steps; these steps achieve $2i+1$ connectivity). These quantities satisfy the recurrence

$$(11)\qquad \begin{array}{rcll} S_1 & \leq & \alpha|OPT|, & \\ S_{i+1} & \leq & \left(\frac{1}{k}+\frac{2i+2}{k^2}\right)|OPT|+\frac{k-1}{k}S_i & \text{for all } (k-1)/2 > i \geq 1. \end{array}$$

This is proved by essentially the same argument as Lemma 7. The differences stem from the fact that the first i steps add a total of i dfs forests and $i+1$ forests of back edges to the solution graph. In more detail, for $j \leq i$ let the jth step add forest F_j and back edges B_j for $j>1$, and forest F_1 and back edges B_0 and B_1 for $j=1$. Lemma 5 and Corollary 6(a) remain valid for B_0 and B_1. (10) becomes

$$(k-2i-1)|B|+|B_0\cap EXT|+\sum_{j=1}^{i}(|B|+|B_j\cap EXT|) \leq |OPT|.$$

Finally, in the last two displayed equations the term $2i$ becomes $(2i+2)$.

It remains to determine α, i.e., we must bound the number of back edges in A in terms of $|OPT|$ (the size of the smallest k-ECSS). We will derive the main theorem using a simple bound for α: [6] gives a short proof that the ear algorithm achieves a 14/9 approximation ratio. We extend that proof as follows.

LEMMA 8. *The 3-ECSS A contains $\le \frac{8}{3k}|OPT|$ back edges.*

Proof. The proof of the 14/9 approximation ratio is based on three lower bounds for the size of the smallest 3-ECSS: the degree lower bound, the carving lower bound, and the component lower bound. All three of these are actually lower bounds on the size of the smallest k-ECSS, which the argument of [6] specializes to $k = 3$. Thus the same argument upper bounds the number of back edges of A in terms of $|OPT|$, the size of the smallest k-ECSS. The modified argument gives the lemma—the only changes are in the algebra of [6, Lemma 4.5]. □

Now use (4c) to solve (11) with $\alpha = 8/(3k)$, $\epsilon = 2$. This gives an upper bound on the number of back edges added by KR3. In addition KR3 adds $(k-1)/2$ dfs forests. The degree lower bound shows they contain a total of $\le \frac{k-1}{k}|OPT|$ edges. Summing the two bounds shows the size of KR3's subgraph is at most

$$\left(1 + \left(1 - \frac{1}{k}\right)^{(k-3)/2}\left(1 - \frac{4/3}{k}\right)\right)|OPT|.$$

This upper bound is illustrated in the row labelled "KRs" in Table 1. As can be seen, the upper bound is less than $1 + \sqrt{1/e}$ (section 2 describes a proof).

THEOREM 9. *The KR algorithm has approximation ratio $\le 1 + \sqrt{1/e}$ on multigraphs with $k > 1$.* □

The bound for odd k displayed above can be improved, for any fixed k, to the row labelled "KR3" in Table 1. This is done by extending the proof of the 3/2 performance ratio for the ear algorithm to show we can take $\alpha = \frac{5}{2k}$ in (11). The extended proof depends on detailed knowledge of [6] and so has been added as an appendix to [6]. Assuming that result we get the following corollary. Note the corollary's bound for $k = 3$ is 3/2 as expected.

COROLLARY 10. *For $k > 1$ the KR algorithm has an approximation ratio at most*

$$\begin{cases} 1 + (1 - \frac{1}{k})^{k/2} & k \text{ even;} \\ 1 + (1 - \frac{1}{k})^{(k-3)/2}(1 - \frac{3/2}{k}) & k \text{ odd.} \end{cases} \qquad \square$$

4. Efficient implementation. When the desired connectivity k is small the implementation of [14] in time $O((kn)^2)$ is efficient. Note that this time bound also applies to our modified version of the algorithm for odd k: That version begins by using the ear algorithm to find a 3-ECSS. The ear algorithm runs in time $O(m\alpha(m, n))$ [6]. Since $O(m\alpha(m, n)) = O(n^2)$ this does not increase the $O((kn)^2)$ time bound. The time bound for the rest of the algorithm is proved exactly as in [14].

The rest of this section concentrates on implementing the KR algorithm efficiently for graphs with large multiplicities. Specifically we assume a weighted graph representation, where each edge is given with an (arbitrarily large) integer multiplicity. We implement the KR algorithm in time dominated by $O(nm)$ maximum flow computations.

The algorithm begins with the initialization just mentioned: If k is odd, the solution graph H is initialized to the 3-ECSS of the ear algorithm; if k is even, H is $\emptyset$. Again the $O(m\alpha(m, n))$ time for the ear algorithm is dominated by the desired time bound. For the rest of the section the parity of k is irrelevant.

The implementation is organized into "phases." Each phase simulates a number of 2-steps, until either the current solution graph H becomes k-edge connected or the multiplicity of one or more edges of $G - H$ decreases to 0 or 1. This implies there are $\le 2m + 1$ phases.

Each phase simulates a sequence of consecutive 2-steps that use the same forest F. The phase consists of a number of "multisteps." A multistep simulates the execution of as many consecutive 2-steps as possible that each add a copy of the same set $F \cup B$.

Fix a 2-step with current graph H and dfs forest F. An edge $xy \in F$ is *critical* if $\lambda(x, y; H \cup F) = h + 1$. We shall see that the critical edges determine the set B. To implement a multistep we need to know when an edge becomes critical. The following lemma answers this question. It also implies several other facts that we need about critical edges.

Take an edge $xy \in F$. The graph HF_{xy} is constructed by starting with H and contracting every connected component of $F - xy$. We use the term "node" to refer to a vertex of HF_{xy}, and node x (node y) refers to the node of HF_{xy} that contains vertex x (vertex y) of G. For a set of edges S, $cov(xy, S)$ denotes the number of edges of S that cover xy. We count each edge according to its multiplicity in S.

To state the lemma, suppose we are executing the KR algorithm (not our implementation). Let H be an h-ECSS of the k-edge connected graph G. Let F be a maximal spanning forest of $G - H$. Take any $c \geq 0$ so $h + 2c \leq k - 2$. Suppose that c consecutive 2-steps enlarge H to the $(h + 2c)$-ECSS $K = H \cup (c \cdot F) \cup B_c$. (In general B_c is a union of a number of different B sets.)

LEMMA 11. *Suppose a $(c + 1)$-st 2-step is to be done, adding a $(c + 1)$-st copy of F. The critical edges in this 2-step are the edges $xy \in F$ satisfying*

$$\lambda(x, y; HF_{xy}) + \mathrm{cov}(xy, B_c) = h + c.$$

Proof. Edge $xy \in F$ is critical for the next 2-step if $\lambda(x, y; K \cup F) = h + 2c + 1$. This means there is a set S of vertices containing x but not y such that

$$d(S, K \cup F) = h + 2c + 1.$$

If $d(S, F) \geq 2$, then $d(S, K \cup F) \geq d(S, H) + (c + 1)d(S, F) \geq h + 2c + 2$. Hence we can assume $d(S, F) \leq 1$. Since xy leaves S, we can assume $d(S, F) = 1$ and xy is the unique edge of F leaving S.

Now we have $d(S, K \cup F) = d(S, H) + (c + 1) + d(S, B_c)$. So the condition for criticality becomes

$$d(S, H) + d(S, B_c) = h + c. \tag{12}$$

Since every edge of B_c joins vertices in the same tree of F, an edge of B_c leaves S if and only if it covers xy, i.e., $d(S, B_c) = \mathrm{cov}(xy, B_c)$. So if S satisfies (12) it must minimize $d(S, H)$. (It is easy to see that any S satisfies (12) with $\geq$, by tracing back to the original criticality condition.) Since xy is the unique edge of F leaving S, S corresponds to a set of nodes in HF_{xy}, and S contains node x but not node y. So (12) is equivalent to the equation of the lemma. □

The special case of the lemma with $c = 0$, $B_c = \emptyset$ is of interest. For instance, consider the following fact.

FACT 1. *Suppose an edge xy is critical at the start of a 2-step. During the bottom-up traversal of the 2-step the quantity $\lambda(x, y; H \cup F \cup B)$ changes from $h + 1$ to $h + 2$ precisely when the first edge covering xy gets added to B.*

To prove Fact 1 use the special case of the lemma. Observe that the first paragraph of the proof of Lemma 11 shows $\lambda(x, y; H \cup F \cup B)$ remains equal to $h + 1$ as long as there is a set S containing x but not y such that xy is the unique edge of F leaving S and no edge of B covers xy.

Fact 1 implies that the set of critical edges C at the start of a 2-step determines the set of back edges B that get added in the 2-step. So we use the notation $B(C)$. The set of "forcing edges" F^* is also determined by C, and we use the notation $F^*(C)$. Note from the bottom-up procedure of a 2-step that $B(C) = B(C')$ for any set C' satisfying $F^* \subseteq C' \subseteq C$.

We need two more facts about critical edges. Consider 2 consecutive 2-steps, where the first 2-step is done for critical edges C and the second 2-step uses the same forest F as the first.

Fact 2. *An edge of C covered by exactly one edge of $B(C)$ is critical for the second 2-step.*

Fact 3. *An edge of $F - C$ that is critical for the second 2-step is not covered by any edge of $B(C)$.*

Facts 2–3 follow from the Lemma 11 by first taking $c = 0$, $B_c = \emptyset$ and then $c = 1$, $B_c = B(C)$.

We can now give the stopping criterion for a multistep. Recall that a multistep simulates the execution of as many consecutive 2-steps as possible that each add a copy of the same set $F \cup B$. In greater detail it simulates 2-steps until either

(a) the number of available copies of some edge of F or B drops to 0, or

(b) some edge of F changes from noncritical to critical.

Let us prove that as long as (a) and (b) do not occur, each 2-step adds the same set $B(C)$. Let $F^* = F^*(C)$. Each edge of F^* is covered by exactly 1 edge of B. Fact 2 shows the set C' of edges that are critical in the next 2-step includes F^*. Assuming (b) does not occur $C' \subseteq C$. This implies $B(C) = B(C')$ as desired.

If (a) occurs the phase ends. (Note that we have described a phase as ending when the number of available copies of some edge drops to 0 or 1, which seems slightly different from (a). The difference is due to the situation where the multiplicity of an edge in $F \cap B$ drops to 1.)

For (b) to occur the new critical edge e must not be covered by B (Fact 3). Now it is easy to see that in the next 2-step the deepest such e enters the set F^* and causes the set B to change.

This justifies the following detailed statement of a multistep: The multistep begins with its set of critical edges C. It determines the set $B = B(C)$. Then it adds the greatest number of copies of $F \cup B$ to H until (a) or (b) occurs. Lemma 11 is used to determine when (b) will occur.

It is not hard to see this gives a polynomial time implementation of a multistep. (Implementation details are given below.) It remains to bound the number of multisteps in a phase. This is nontrivial because in a sequence of consecutive 2-steps with the same forest F, a given edge of F can change from critical to noncritical and back again many times.

It is convenient to extend some tree terminology from vertices to tree edges. An *edge-ancestor* of a vertex v is a tree edge whose deeper vertex is an ancestor of v. An edge-ancestor of a tree edge e is an edge-ancestor of the deeper vertex of e. If e is a tree edge then *depth*(e) is the depth of the deeper vertex of e. We start by describing how F^* changes when a new edge becomes critical.

Lemma 12. *Let C be a set of critical edges. Let c be a tree edge that is not covered by $B(C)$ but is covered by some back edge. Then for some edge f covering c either*

(a) *$F^*(C \cup c) = F^*(C) + c$ and $B(C \cup c) = B(C) + f$, or*

(b) *for some set $D \subseteq C \cup c$, some proper edge-ancestor $\underline{a}$ of c, and some edge e covering $\underline{a}$, $F^*(D) = F^*(C) - a + c$ and $B(D) = B(C) - e + f$.*

Proof. Let f be a back edge that covers c and is directed to the shallowest possible vertex, say w. Let a be the deepest edge-ancestor of c belonging to $F^*(C)$, if such exists. We will show that (b) holds if a exists and (a) holds if it does not. If a exists let $e \in B(C)$ be the edge covering a and let v be the head of e. Since a is an edge-ancestor of c, v is an ancestor of w. Let P_{wv} consist of all edges on the tree path from w to v. If v does not exist then take $v = w$ and $P_{wv} = \emptyset$. Let

$$D = C - P_{vw} + c.$$

Note that $D = C \cup c$ if a does not exist.

We will execute the bottom-up procedure to construct both $B(C)$ and $B(D)$. The desired relations (a)–(b) result from the fact that the two executions differ only in 1 step.

Let F_- consist of all edges of F except the edge-ancestors of c. Clearly $C \cap F_- = D \cap F_-$. Hence the bottom-up procedure works the same for C and D on F_-, placing the same edges of F_- into both $F^*(C)$ and $F^*(D)$ and the same back edges into $B(C)$ and $B(D)$.

The next step of the bottom-up procedure adds f to $B(D)$ and c to $F^*(D)$. If a does not exist then both procedures are done, and part (a) of the lemma holds. Suppose a exists. The bottom-up procedure adds e to $B(C)$ and a to $F^*(C)$.

The remaining steps of both executions are exactly the same. This follows since the remaining uncovered edges of C are edge-ancestors of v, by definition. The remaining uncovered edges of D are edge-ancestors of w, and since $D \cap P_{wv} = \emptyset$, they are edge-ancestors of v. C and D contain exactly the same edge-ancestors of v. So the remaining steps of both executions are the same, and we have verified (b) of the lemma. □

Consider the potential function

$$\Phi = \sum\{\text{depth}(e) : e \in F^*\}.$$

For Φ we compute depths in the forest F. We will prove that every multistep in a phase except possibly the last increases Φ by at least 1. Since $\Phi \leq n^2$, this implies that a phase has $O(n^2)$ multisteps.

If C is a set of critical edges, let $\Phi(C)$ denote the above expression with $F^* = F^*(C)$. When we apply Lemma 12 it is convenient to unify the two cases by taking D to be $C \cup c$ in case (a). Observe that $\Phi(D) > \Phi(C)$ for both cases (a) and (b) of Lemma 12 (since $\Phi(D) = \Phi(C) + \text{depth}(c) - \text{depth}(a)$, where we take $\text{depth}(a) = 0$ in (a)).

LEMMA 13. *Every multistep in a phase except the first or last increases Φ.*

Proof. Consider a multistep that is not the first or last of its phase. Let C_0 be the set of edges that were critical in the previous multistep and remain critical at the start of the current multistep. Let C_1 be the set of edges that have changed from noncritical to critical. Fact 2 implies that B of the previous multistep equals $B(C_0)$. Fact 3 implies no edge of C_1 is covered by $B(C_0)$.

We will construct sets $D_i \subseteq C_0 \cup C_1$, $i = 0, \ldots, I$ for some $I \geq 1$, with $\Phi(D_i) > \Phi(D_{i-1})$, $D_0 = C_0$, and $D_I = C_0 \cup C_1$. Clearly this implies the lemma. We will also have each D_i maximal in the sense that D_i contains every edge of $C_0 \cup C_1$ that is covered by $B(D_i)$. Clearly D_0 is maximal.

Supposing D_{i-1} has been constructed, construct D_i as follows. Take an edge $c \in C_0 \cup C_1 - D_{i-1}$ that has maximum possible depth. (For $i = 1$, $C_1 - C_0 \neq \emptyset$ since this is not the last multistep of the phase.) By maximality c is not covered by $B(D_{i-1})$. Apply Lemma 12 to $C \equiv D_{i-1}$ and c. The lemma gives set $D \equiv D_i$ with $D_i \subseteq C \cup c \subseteq C_0 \cup C_1$ and $\Phi(D_i) > \Phi(D_{i-1})$. We can enlarge D_i so it contains all edges of $C_0 \cup C_1$ that are covered by $B(D_i)$, without changing $F^*(D_i)$ or $B(D_i)$. (For edge f this depends on the choice of c to have maximum depth. For other edges, which belong to D_{i-1}, this depends on the maximality of D_{i-1}.) Now D_i is maximal. If now $D_i = C_0 \cup C_1$ then take $I \equiv i$ and we are done. This eventually occurs since Φ always increases. □

It is easy to see that the lemma guarantees each phase runs in polynomial time and we have a polynomial-time algorithm. We now show the time for a phase is dominated by $O(n)$ maximum flow computations. We first assume flows are computed by the algorithm of [16] which uses time $O(nm \log_b n)$ for $b = m/(n \log n) + 2$.

A phase starts by doing a dfs to determine F. It then computes two sets of values that remain constant during the phase: first, the values $\lambda(x, y; HF_{xy})$, $xy \in F$. These values do not change since F does not change. The values are computed using $O(n)$ flow computations, each on a graph of $\leq n$ vertices and $\leq m$ edges.

The second set of values are the *lowpoint* values of the vertices of F [14, 19]. These values do not change since the set of nontree edges does not change. The *lowpoint* values are computed in $O(m)$ time. This enables each multistep to determine the new set B in a bottom-up traversal of F in $O(n)$ time.

The phase also maintains the values $\mathrm{cov}(xy, B_c)$ for each $xy \in F$. Here B_c is the set of nontree edges added so far in the phase. Initially all cov values are 0.

Now we show that each multistep can be executed in $O(n)$ time. Use Lemma 11 to determine the set of critical edges C for this multistep. All quantities needed have already been computed, so this takes $O(n)$ time. Compute $B = B(C)$ in $O(n)$ time using *lowpoint* values.

Determine the greatest number of 2-steps that can be executed, as follows. Compute the values $\mathrm{cov}(xy, B)$ for $xy \in F$ and the new B in time $O(n)$. Edges with $\mathrm{cov}(xy, B) = 0$ are candidates for becoming critical. For each such xy the equation of Lemma 11 gives the value of c at which xy becomes critical. Let c_0 be the least such value (if it exists).

Let the current solution graph have edge connectivity $h + 2c$. Let a be the fewest number of available copies of an edge of F or B. The multistep adds $d = \min\{c_0, a, (k-(h+2c))/2\}$ copies of $F \cup B$. Finally the multistep increases each value $\mathrm{cov}(xy, B_c)$ by $d \times \mathrm{cov}(xy, B)$.

Lemma 13 shows the total time for all multisteps in a phase is $O(n^3)$. This is within the time bound for a phase.

Theorem 14. *The KR algorithm can be implemented on multigraphs in time* $O((nm)^2 \log_b n)$ *for* $b = m/(n \log n) + 2$. □

Now we assume that maximum flows are computed by the algorithm of [9]. If all capacities are integers $\leq k$ this algorithm runs in time $O(\mu m \log(n^2/m) \log k)$ for $\mu = \min\{m^{1/2}, n^{2/3}\}$.

A phase starts as before, computing F, the values $\lambda(x, y; HF_{xy})$ for $xy \in F$, and all *lowpoint* values. In the flow computations we can assume all capacities in HF_{xy} are integers $\leq k$, since flow values larger than k are irrelevant. Hence the desired time bound for computing a flow applies.

We must speed up the implementation of a multistep. We achieve total time

$O((n \log n)^2)$ for all multisteps in a phase. Since this quantity is $O(\mu nm)$ it suffices for our desired time bound. The approach is based on a stronger version of Lemma 13. Recall that Lemma 12 shows how to make a new edge c critical, by adding an edge to both F^* and B and perhaps dropping an edge from both of these sets. Call this change to F^* and B an *edge swap*. The proof of Lemma 13 shows how to update the set B from one multistep to the next, using a total of $O(n^2)$ edge swaps in the entire phase. Our algorithm updates B using this procedure. We will show that each edge swap can be done in time $O(\log^2 n)$. We sketch the implementation, leaving some details to the interested reader.

We first describe the mechanism for determining critical edges. For $xy \in F$ define $t(xy) = \min\{\lambda(x, y; HF_{xy}) + \text{cov}(xy, B_c), k\}$. Lemma 11 shows that if xy is not covered by the current B set, $t(xy) < k$, and every following 2-step adds the current B, then xy becomes critical after step $t(xy) - h$, where we have indexed the 2-steps of the current phase starting at 1.

Define a key $T(xy)$ for each $xy \in F$ by

$$T(xy) = K^2\text{cov}(xy, B) + Kt'(xy) + (n - \text{depth}(xy)).$$

Here $K = \max\{k, n\}$. B denotes the current set B (even in the middle of the bottom-up procedure). $t'(xy)$ is the quantity $t(xy)$ excluding the contribution of all current edges B to $\text{cov}(xy, B_c)$. Observe that an edge xy that is not covered by B has $T(xy) = Kt(xy) + (n - \text{depth}(xy))$. So it is easy to see that the edge xy with smallest key $T(xy)$ is the next edge to turn critical, and if there is a tie for this edge, xy is as deep as possible. (The latter is needed since we follow the proof of Lemma 13 to update B.)

Our data structure is based on the heavy path decomposition [20] of the forest F. Recall this notion gives a partition of the edges of F into "heavy paths." Each edge of a given heavy path has, to within a factor of 2, the same number of vertex descendants in F. The path from any vertex to the root of its tree in F intersects at most $\log n$ heavy paths.

The edges of each heavy path are stored in a binary search tree, where symmetric order is the same as order in the heavy path. Each node of the search tree keeps track of the smallest key $T(xy)$ in its subtree, and each node has a displacement quantity that is added to all the keys in its subtree. (This enables all keys in a subtree to be increased by the same value in time $O(1)$.) There is a priority queue Q that contains the edge of smallest key from each heavy path. Finally, each edge of F^* is marked in the heavy path decomposition, so that the deepest edge-ancestor in F^* of a given tree edge can be found in $O(\log^2 n)$ time.

At the start of a multistep we find the next new critical edge using the queue Q in time $O(\log n)$. We update the set B by following the procedure of Lemma 13. Each swap is done in $O(\log^2 n)$ time. The number of 2-steps to be simulated is computed as before. We achieve time $O((n \log n)^2)$ for a phase as desired.

COROLLARY 15. *The KR algorithm can be implemented on multigraphs in time* $O(\mu nm^2 \log(n^2/m) \log k)$ *for* $\mu = \min\{m^{1/2}, n^{2/3}\}$. □

Appendix. Lower bound examples.

A.1. The Cheriyan–Thurimella algorithm on multigraphs.

Throughout this section take any fixed $k \geq 2$. The algorithm of Cheriyan and Thurimella [3] consists of two steps: First take M to be a minimum cardinality set of edges in which every vertex has degree $\geq k$. Next take C to be a minimal set of edges that makes $M \cup C$ k-edge connected. Return the k-ECSS $M \cup C$.

We will exhibit multigraphs that have $|M \cup C| \geq (2-\epsilon)|OPT|$, for any $\epsilon > 0$. On the other hand we will prove the approximation ratio is always ≤ 2. This upper bound holds even for the weaker version of the algorithm where M is chosen to be a minimal set rather than a minimum cardinality set. Cheriyan and Thurimella [3] prove much stronger upper bounds when the graph is restricted to be simple.

To prove the upper bound suppose that the edges of $M \cup C$ can be partitioned into k forests. Then using the degree lower bound we get $|M \cup C| \leq k(n-1) < kn \leq 2|OPT|$ as desired. Hence our upper bound follows from this lemma.

Lemma 16. *The edges of $M \cup C$ can be partitioned into k spanning forests.*

Proof. We first partition M into spanning forests $F_1, \ldots, F_k$. To do so repeatedly remove a maximal spanning forest from M until k forests are obtained. Let $N = \cup_{i=1}^{k} F_i$. We claim $N = M$. If not take an edge $xy \in M - N$. Vertices x and y are in the same tree in each forest F_i, by maximality. Hence $d(x, N), d(y, N) \geq k$. But now edge xy contradicts the minimality of M.

Next, take each edge $e \in C$ and add it to a forest F_i where it joins 2 different trees. We claim this is always possible. If not suppose $e = xy \in C$ cannot be added to any forest. Thus each forest F_i contains an xy-path. This implies $\lambda(x, y; M \cup C - e) \geq k$. But this contradicts the minimality of C. □

We turn to the example graphs. Let C_n denote the cycle on n vertices. Index its vertices from 0 to $n-1$ in cyclic order.

First, suppose k is even. Let $G = k \cdot C_n$. For simplicity assume n is even (a similar construction works if n is odd). $\frac{k}{2} \cdot C_n$ is a k-ECSS of G with exactly $kn/2$ edges. Since this matches the degree lower bound this graph can be taken as OPT.

Execute the approximation algorithm on G as follows. In the first step choose $M = k \cdot \{(2i, 2i+1) : i = 0, \ldots, n/2-1\}$. (This is valid since every vertex has degree exactly k.) In the second step choose $C = k \cdot \{(2i-1, 2i) : i = 1, \ldots, n/2-1\}$. (This is valid since it makes $M \cup C$ a minimal k-edge connected graph.) We have $|M \cup C| = kn/2 + k(n/2-1) = k(n-1)$. Thus $|M \cup C|/|OPT| = 2(n-1)/n$ which approaches 2 as $n \to \infty$.

A similar example works if k is odd and ≥ 3. The graph is $k \cdot C_n$ plus the matching

$$A = \{(2i, 2i+3) : i = 0, \ldots, n/2-1\}.$$

Here we assume n is even and ≥ 4, and addition is modulo n. It is easy to see that for $k = 1$ this graph is 3-edge connected. Gabow ([6] also uses this 3-edge connected graph.) The subgraph $\frac{k-1}{2} \cdot C_n \cup A$ is k-edge connected (since $C_n \cup A$ is 3-edge connected and $\frac{k-3}{2} \cdot C_n$ is $k-3$-edge connected). Each vertex has degree k so again we get $|OPT| = kn/2$. The approximation algorithm can choose exactly the same sets M, C as before, so again the approximation ratio approaches 2.

A.2. The KR algorithm. Khuller and Vishkin give an example showing their algorithm for approximating the smallest 2-ECSS can have performance ratio approaching 3/2 [15]. The example is robust—it remains valid even when we add a postprocessing step that ensures the algorithm returns a minimal k-ECSS. We adapt this example to show the same performance characteristics for the KR algorithm.

First, assume k is even. The graph G consists of two spanning subgraphs, the first being OPT and the second being the subgraph returned by KR, say ALG. OPT is $\frac{k}{2} \cdot C_n$, where as before the vertices are indexed from 0 to $n-1$ in cyclic order and n is even for convenience. Notice that no two vertices of the same parity are adjacent in OPT. (We identify a vertex and its index.) We will maintain this property for the odd vertices.

The KR algorithm operates in $\frac{k}{2}$ 2-steps. Each 2-step will add the same set of edges, specifically a dfs tree T and a set of back edges B. So $ALG = \frac{k}{2} \cdot (T \cup B)$. The tree T consists of a path spanning the even vertices and leading to leaves that are the odd vertices. More precisely T has root 0, the path is $0, 2, 4, \ldots, 2n-2$ and for $i = 1, \ldots, n/2$ the leaf $2i-1$ is on the edge $(2i-1, 2n-2)$. The set B contains a back edge from each leaf to the root, i.e., $(2i-1, 0)$, $i = 1, \ldots, n/2$. Observe that as mentioned, $G = OPT \cup ALG$ does not have any edge joining two odd vertices.

Let us verify that the KR algorithm actually returns the subgraph ALG of G. Suppose inductively that the algorithm has constructed the h-ECSS $H = \frac{h}{2} \cdot (T \cup B)$ and we execute a 2-step. T is a valid dfs tree of $G - H$ because each leaf $2i-1$ is only adjacent to even vertices, which occur as ancestors of $2i-1$ in T. Each leaf has degree exactly $h+1$ in $H \cup T$. Hence it is easy to see that for each leaf $2i-1$ the algorithm adds the back edge $(2i-1, 0)$ to B. Since $T \cup B$ is 2-edge connected, adding it increases the connectivity to $h+2$ as desired.

Finally, note that $|T \cup B| = (n-1) + n/2 = 3n/2 - 1$. Hence $|ALG|/|OPT| = (k/2)(3n/2-1)/(kn/2) = 3/2 - 1/n$ which approaches $3/2$ as $n \to \infty$. Furthermore, ALG is a minimal k-ECSS. This follows from the fact that every vertex except 0 and $2n-2$ has degree exactly k.

A similar example gives the same performance ratio for the original KR algorithm when k is odd. We can take $OPT = \frac{k-1}{2} \cdot C_n \cup A$ for the matching A defined in the previous section. Again $|OPT| = kn/2$, and no edge joins two odd vertices. Take $ALG = \frac{k-1}{2} \cdot (T \cup B) \cup T$. The KR algorithm constructs the $(k-1)$-edge connected graph $\frac{k-1}{2} \cdot (T \cup B)$ just as before. Then it adds T to achieve k-edge connectivity. Since $|ALG| = \frac{k+1}{2}(n-1) + \frac{k-1}{2}\frac{n}{2} \geq \frac{k}{2}(3n/2-1)$, the approximation ratio approaches a quantity $\geq 3/2$ as before.

Acknowledgment. We are indebted to an anonymous referee for pointing out the linear programming approach of reference [10].

REFERENCES

[1] M. Cai, *The number of vertices of degree k in a minimally k-edge-connected graph*, J. Combin. Theory Ser. B, 58 (1993), pp. 225–239.

[2] J. Cheriyan, A. Sebö, and Z. Szigeti, *Improving on the* 1.5*-approximation of a smallest* 2*-edge connected spanning subgraph*, SIAM J. Discrete Math., 14 (2001), pp. 170–180.

[3] J. Cheriyan and R. Thurimella, *Approximating minimum-size k-connected spanning subgraphs via matching*, SIAM J. Comput., 30 (2000), pp. 528–560.

[4] C. G. Fernandes, *A better approximation ratio for the minimum size k-edge-connected spanning subgraph problem*, J. Algorithms, 28 (1998), pp. 105–124.

[5] A. Frank, *Submodular functions in graph theory*, Discrete Math., 111 (1993), pp. 231–243.

[6] H. N. Gabow, *An ear decomposition approach to approximating the smallest* 3*-edge connected spanning subgraph of a multigraph*, SIAM J. Discrete Math., 18 (2004), pp. 41–70.

[7] H. N. Gabow, *On the Difficulty of k-Connected Spanning Subgraph Problems*, unpublished notes.

[8] H. N. Gabow, M. X. Goemans, É. Tardos, and D. P. Williamson, *Approximating the smallest k-edge connected spanning subgraph by LP-rounding*, in Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2005, pp. 562–571.

[9] A. V. Goldberg and S. Rao, *Beyond the flow decomposition barrier*, J. Assoc. Comput. Mach., 45 (1998), pp. 783–797.

[10] M. X. Goemans, É. Tardos, and D. P. Williamson, *private communication*, 1994, cited in D. R. Karger, *Random Sampling in Graph Optimization Problems*, Ph.D. dissertation, Department of Computer Science, Stanford University, Stanford, CA, 1995, p. 205.

[11] R. Jothi, B. Raghavachari, and S. Varadarajan, *A* 5/4*-approximation algorithm for minimum* 2*-edge-connectivity*, in Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2003, pp. 725–734.

[12] D. R. Karger, *Random sampling in cut, flow, and network design problems*, Math. Oper. Res., 24 (1999), pp. 383–413.

[13] S. Khuller, *Approximation algorithms for finding highly connected subgraphs*, in Approximation Algorithms for NP-hard Problems, D. S. Hochbaum, ed., PWS Publishing, Boston, 1997, pp. 236–265.

[14] S. Khuller and B. Raghavachari, *Improved approximation algorithms for uniform connectivity problems*, J. Algorithms, 21 (1996), pp. 434–450.

[15] S. Khuller and U. Vishkin, *Biconnectivity approximations and graph carvings*, J. Assoc. Comput. Mach., 41 (1994), pp. 214–235.

[16] V. King, S. Rao, and R. Tarjan, *A faster deterministic maximum flow algorithm*, J. Algorithms, 17 (1994), pp. 447–474.

[17] P. Krysta and V. S. Anil Kumar, *Approximation algorithms for minimum size* 2*-connectivity problems*, in Proceedings of the 18th International Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Comput. Sci. 2010, Springer-Verlag, Berlin, 2001, pp. 431–442.

[18] H. Nagamochi and T. Ibaraki, *Computing edge-connectivity in multigraphs and capacitated graphs*, SIAM J. Discrete Math., 5 (1992), pp. 54–66.

[19] R. E. Tarjan, *Depth-first search and linear graph algorithms*, SIAM J. Comput., 1 (1972), pp. 146–160.

[20] R. E. Tarjan, *Applications of path compression on balanced trees*, J. Assoc. Comput. Mach., 26 (1979), pp. 690–715.

[21] V. V. Vazirani, *Approximation Algorithms*, Springer-Verlag, Berlin, 2001.

[22] S. Vempala and A. Vetta, *Factor* 4/3 *approximations for minimum* 2*-connected subgraphs*, in Approximation Algorithms for Combinatorial Optimization, K. Jansen and S. Khuller, eds., Lecture Notes in Comput. Sci. 1913, Springer-Verlag, Berlin, 2000, pp. 262–273.

SIAM J. Discrete Math.
Vol. 19, No. 1, pp. 19–45

BIMODALITY AND PHASE TRANSITIONS IN THE PROFILE VARIANCE OF RANDOM BINARY SEARCH TREES*

MICHAEL DRMOTA[†] AND HSIEN-KUEI HWANG[‡]

Abstract. We show that the variances of the profile (number of nodes at each level) of random binary search trees undergoes asymptotically four phase transitions and exhibits a bimodal or "two-humped" behavior, in contrast to the unimodality of the expected value of the profiles. Precise asymptotic approximations are derived. The same types of phenomena also hold for the profile of random recursive trees.

Key words. binary search trees, asymptotic bimodality, profile, Bessel functions, Stirling numbers of the first kind, singularity analysis, saddle-point method

AMS subject classifications. 60C05, 68P05, 68P10

DOI. 10.1137/S0895480104440134

1. Introduction.

Profile (sequence of numbers of nodes having the same distance to the root) is an informative shape characteristic of trees. It is directly related to the total path length (the sum of the distances of all nodes to the root) and depth (the distance of a random node to the root) on the one hand, and can be used to derive effective bounds for the height and width on the other hand. In terms of branching process language, profiles correspond to the number of descendants in each generation; they also have more concrete algorithmic interpretations such as breadth-first search and applications; see Devroye and Robson (1995), Louchard and Szpankowski (1995), Chern and Hwang (2001). In this paper we study the variance of the profile in random binary search trees (abbreviated as BSTs). Part of our aim is to clarify Figure 1.1 by more precise mathematical terms.

Binary search trees. A BST $\mathcal{T}$ is a binary tree constructed from a given sequence of keys, say $\mathcal{A} := \{a_1, \dots, a_n\}$, as follows. If $n = 0$, then $\mathcal{T}$ is empty and, for convenience, we regard $\mathcal{T}$ as consisting of only a node called *external node*. If $n \geq 1$, then the first key a_1 is placed at the root (called an *internal node*). The remaining keys are compared successively to the root key and are directed to the left (or right) branch if they are smaller (or larger), and keys directed to the same branch are constructed recursively as a BST. By construction, a query operation like "$x \in \mathcal{T}$?" can be easily carried out in BSTs, thus the name.

BSTs are one of the simplest and widely used data structures in computer algorithms. They also appeared, under different guises, in other contexts such as branching processes, population genetics, diffusion models, and evolutionary trees; see Aldous and Shields (1988), Aldous (1996), Barlow, Pemantle, and Perkins (1997), Majumdar and Krapivsky (2003). The large number of diverse extensions and variants add significantly to their importance in practice, algorithm design, and theory.

*Received by the editors January 22, 2004; accepted for publication (in revised form) September 8, 2004; published electronically June 3, 2005.

http://www.siam.org/journals/sidma/19-1/44013.html

[†] Institut für Diskrete Mathematik und Geometrie, Technische Universität Wien, Wiedner Hauptstrasse 8-10/118, 1040 Wien, Austria (michael.drmota@tuwien.ac.at).

[‡] Institute of Statistical Science, Academia Sinica, Taipei 115, Taiwan (hkhwang@stat.sinica.edu.tw). This author's research was partially supported by a Research Award from the Alexander von Humboldt Foundation.

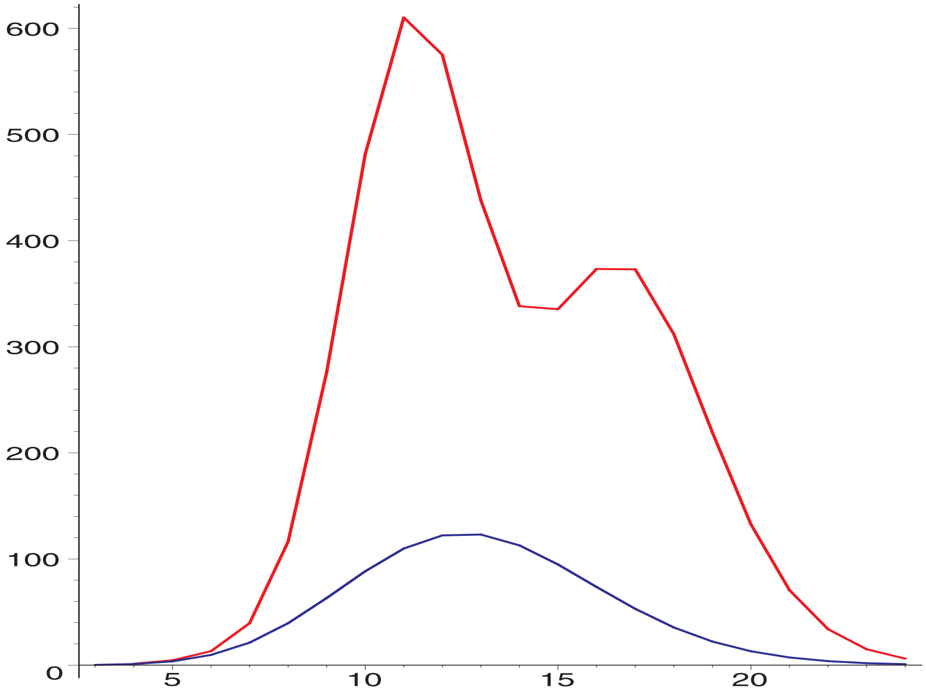


FIG. 1.1. *Profiles of BSTs: spline curves for the exact mean (the unimodel curve) and the exact variance (the bimodal curve) of the number $X_{1000,k}$ of external nodes at level k in random binary search trees of* 1000 *nodes.*

Random BSTs. Assume that the given input is a finite sequence of independent and identically distributed random variables with a common continuous distribution. The BST constructed from this random sequence is called a *random BST*. Since only the rank and the order of the keys are relevant, an equivalent model is to assume that the input is a random permutation when all $n!$ permutations of n elements are equally likely.

Many properties of random BSTs have been studied in the literature; see Gonnet and Baeza-Yates (1990), Mahmoud (1992), Knuth (1998), Devroye (2003), Hwang and Neininger (2002), Chauvin, Drmota, and Jabbour-Hattab (2001), Chauvin et al. (2005), Chauvin and Rouault (2004), and Fuchs, Hwang, and Neininger (2004) for more information.

Profiles of random BSTs. We are concerned with the random variables $X_{n,k}$, defined to be the number of external nodes at level k (the root being at level 0) in a random BST of n nodes. It is known that

$$\mathbb{E}(X_{n,k}) = \frac{2^k}{n!}\, s(n,k) \qquad (0 \le k \le n), \tag{1.1}$$

where the $s(n,k)$'s denote the signless Stirling numbers of the first kind,

$$\sum_{0\le k\le n} s(n,k)w^k = w^{\overline{n}} \qquad (n \ge 0),$$

with $w^{\overline{n}}$ denoting the rising factorial $w^{\overline{n}} := \prod_{0\le j<n}(w+j)$; see Lynch (1965), Knuth (1998), Brown and Shubert (1984), Mahmoud and Pittel (1984), Pittel (1984),

Louchard (1987), Devroye (1988). Thus the asymptotic behaviors of $\mathbb{E}(X_{n,k})$ can be derived from known results for Stirling numbers $s(n,k)$; see Hwang (1995), Temme (1993).

In particular, the asymptotic behaviors of $\mathbb{E}(X_{n,k})$ for varying k are well approximated by a normal distribution, with mode near $k \approx 2\log n$; see Jabbour-Hattab (2001) and Chauvin et al. (2005) for more precise properties. Thus the profile of random BSTs is generally described by the fig-like shape ◇. Note that the sequence $\{\mathbb{E}(X_{n,k})\}_k$ for fixed n is unimodal, by the simple fact that the generating polynomial $\sum_k \mathbb{E}(X_{n,k})w^k$ has only real zeros; see Comtet (1974), Hammersley (1951).

Known results beyond mean. Almost sure convergence of $X_{n,k}/\mathbb{E}(X_{n,k})$ and other type of results are derived in Chauvin, Drmota, and Jabbour-Hattab (2001), Jabbour-Hattab (2001); see also the recent papers by Chauvin et al. (2003), Chauvin and Rouault (2004). Pittel (1984) derived the expression

$$\mathbb{E}(X_{n,k}^2) = \frac{2^k}{n!} \sum_{1\le t\le n} \frac{1}{(2\pi i)^2} \oint\!\!\oint \frac{(\sqrt{8}x/y-1)^{\overline{t-1}}(x^2+t)^{\overline{n-t}}}{yx^{2k-1}\sqrt{1-y^2}}\, \mathrm{d}x\, \mathrm{d}y,$$

and then showed that

$$\mathbb{E}(X_{n,k}^2) = O((\log n)^{3/2} n^{2(\alpha-\alpha\log(\alpha/2)-1)}) \qquad (2-\sqrt{2}\le\alpha\le 2+\sqrt{2}),$$

where, here and throughout this paper, $\alpha := k/\log n$.

Global description of the phase transitions. The aim of this paper is to derive more precise asymptotic approximations to the variance $\mathbb{V}(X_{n,k})$ for all ranges of interest. We show that the asymptotic behavior of $\mathbb{V}(X_{n,k})$ exhibits phase transitions at the four points $\alpha = 3\pm 2\sqrt{2}$ and $\alpha = 2\pm\sqrt{2}$ (not viewable from Figure 1.1 though). The rough picture of $\mathbb{V}(X_{n,k})$ is as follows; see Theorem 2 for a more precise statement.

– When α is small or large, more precisely, $0\le\alpha\le 3-2\sqrt{2}-\varepsilon$ or $\alpha\ge 3+2\sqrt{2}+\varepsilon$, then the variance is of the same order as the mean

$$\mathbb{V}(X_{n,k}) \sim \mathbb{E}(X_{n,k}^2) \asymp \mathbb{E}(X_{n,k}),$$

where $a_n \asymp b_n$ if both $a_n = O(b_n)$ and $b_n = O(a_n)$ hold.

– When α lies in the middle range, namely, $2-\sqrt{2}+\varepsilon\le\alpha\le 2+\sqrt{2}-\varepsilon$, then the variance is of the order of $(\mathbb{E}(X_{n,k}))^2$,

$$\mathbb{V}(X_{n,k}) \sim \varphi(\alpha)(\mathbb{E}(X_{n,k}))^2, \tag{1.2}$$

where

$$\varphi(\alpha) := \frac{\Gamma(\alpha)^2\alpha^2(2\alpha-1)}{\Gamma(2\alpha)(4\alpha-\alpha^2-2)} - 1, \tag{1.3}$$

Γ being the gamma function.

– When α lies in the two intermediate ranges $3-2\sqrt{2}+\varepsilon\le\alpha\le 2-\sqrt{2}-\varepsilon$ and $2+\sqrt{2}+\varepsilon\le\alpha\le 3+2\sqrt{2}-\varepsilon$, then the variance is larger in order than the mean and the mean square:

$$\mathbb{E}(X_{n,k}), (\mathbb{E}(X_{n,k}))^2 \ll \mathbb{V}(X_{n,k}) \sim \mathbb{E}(X_{n,k}^2).$$

Note that $\mathbb{E}(X_{n,k}) = o(1)$ for $\alpha < \alpha_-$ and $\alpha > \alpha_+$, where $\alpha_- \approx 0.37336\ldots$ and $\alpha_+ \approx 4.31107\ldots$ are the two zeros of the equation $e^{(z-1)/z} = z/2$ (sometimes called the binary search tree constants; see Finch (2003, section 5.13)).

To bridge the asymptotic estimates in neighboring ranges, we need more uniform estimates. We show that the transition is well dictated by a *parabolic cylinder function* when α crosses $3 \pm 2\sqrt{2}$ and by a *normal distribution function* when α crosses the other two transitional points.

The valley. The approximation (1.2) in the middle range is insufficient for describing the behaviors of the variance when $\alpha \approx 2$ since $\varphi(2) = \varphi'(2) = 0$. More precise approximations are thus needed, and we derive an asymptotic expansion for $\mathbb{V}(X_{n,k})$ in the middle range. In particular, the visible valley in Figure 1.1 is roughly due to the estimates

$$\begin{aligned}\mathbb{V}(X_{n,\lfloor 2\log n+O(1)\rfloor}) &\asymp \frac{n^2}{(\log n)^3},\\ \mathbb{V}\left(X_{n,\left\lfloor 2\log n\pm\sqrt{2\log n}\right\rfloor}\right) &\asymp \frac{n^2}{(\log n)^2}.\end{aligned}$$

Indeed, we show that

$$\max_{k\ge 0}\mathbb{V}(X_{n,k}) \sim \frac{21-2\pi^2}{24\pi e}\cdot\frac{n^2}{(\log n)^2}.$$

See section 5 for a more precise description of the valley, including an explanation of why the left "hump" is higher than the right one.

Numerically, the first valley for $\mathbb{V}(X_{n,k})$ appears at $n = 357$.

A "false valley." While the valley near $2\log n$ may be quite expected (see Chauvin, Drmota, and Jabbour-Hattab (2001) and Chauvin et al. (2005)), the function $\varphi(\alpha)$ also satisfies $\varphi(1) = \varphi'(1) = 0$, suggesting that there may be a second valley near $\alpha \sim 1$. We show that this is indeed a "false valley" since the decrease of the variance in the logarithmic term is well "smoothed out" by other larger factors; see Corollary 5.

Why the valley? Structurally, the valley for the variance near $k = 2\log n + O(\sqrt{\log n})$ indicates that there is a better concentration of external nodes near these levels, and indeed almost all external nodes lie at these levels, each level having about $n/\sqrt{\log n}$ nodes; see also Chauvin, Drmota, and Jabbour-Hattab (2001). Similarly, the "false valley" near $k = \log n + O(\sqrt{\log n})$ may be ascribable to the structural change of number of internal nodes near there.

Methodology. Our approach is mostly analytic and relies on integral representations for the second moments. The basic idea is to consider the bivariate generating function, say $F_2(z, w)$ of $E(X_{n,k}(X_{n,k} - 1))$, which satisfies a differential equation of the first order. Solving the differential equation yields an integral representation for F_2, from which we apply Cauchy's integral expression and complex-analytic tools, including singularity analysis, saddlepoint method, and some uniform asymptotic methods (for handling the coalescence of a saddlepoint and an algebraic singularity). The approach is of some generality and may be applied to other log-class of trees (of which BST is a prototype); see Bergeron, Flajolet, and Salvy (1992), Devroye (1999). For a different, elementary approach, see Fuchs, Hwang, and Neininger (2004).

Universality? The above interesting phenomena naturally suggest the question: Are the phase transitions and bimodality unique for BSTs? Or is there some sort of universality for such phenomena? We will briefly examine recursive trees in section 7, and show that the profile variance also exhibits a bimodality near $\log n$ and two phase transitions. Similar behaviors are expected for other classes of trees like m-ary search trees, fringe-balanced BSTs (see Devroye (1999)), but the precise description and general prediction are expected to be more involved.

Limit distribution. It is known that (see Chauvin et al. (2005))

$$\frac{X_{n,k}}{\mathbb{E}(X_{n,k})} \to X_{\bar{\alpha}/2} \qquad (\alpha_- < \alpha < \alpha_+),$$

almost surely, where $\bar{\alpha} := \lim_n k/\log n$, $X_z \overset{d}{=} zU^{2z-1}X_z + z(1-U)^{2z-1}X'_z$, U being uniformly distributed in the unit interval and $X'_z \overset{d}{=} X_z$; see also Jabbour-Hattab (2001). Note that $X_{1/2} = X_1 = 1$. The limit distributions of $(X_{n,k} - \mathbb{E}(X_{n,k}))/\sqrt{\mathbb{V}(X_{n,k})}$ in the two special cases $\alpha \sim 1, 2$ were recently derived in Fuchs, Hwang, and Neininger (2004), as well as the somewhat unexpected result that $(X_{n,k} - \mathbb{E}(X_{n,k}))/\sqrt{\mathbb{V}(X_{n,k})}$ does not converge to a fixed limit law when $k = 2\log n + O(1)$.

Profiles of another class of trees (which we may roughly term as "$\sqrt{n}$-class," in contrast to our "$\log n$-class" of trees) have received much recent interests and are now well clarified (see Aldous (1991), Drmota and Gittenberger (1997), Pitman (1999), Kersting (1998)), but many properties of the profiles for the $\log n$-class of trees remain very challenging; see our recent progress in Fuchs, Hwang, and Neininger (2004).

Outline of the paper. This paper is organized as follows. We first derive the basic recurrence for the profiles in the next section, and then the solution to the generating function of mth moments. In particular, an exact solution for the second factorial moment is given. We then state our main results on phase transitions and bimodality in section 3. Proofs are given in later sections, and recursive trees are briefly examined in section 7.

2. Generating functions and integral representations. We give here a self-contained approach to computing the moments of $X_{n,k}$. Define the bivariate generating function

$$P_k(z,y) := \sum_{n\ge 0} \mathbb{E}(y^{X_{n,k}})z^n \qquad (k \ge 0).$$

Then, by the recursive construction,

$$X_{n,k} \overset{d}{=} X_{I_n,k-1} + X^*_{n-I_n-1,k-1},$$

where I_n is uniformly distributed in $\{0,1,\dots,n-1\}$, $(I_n), (X_{n,k}), (X^*_{n,k})$ are independent, and $X^*_{n,k} \overset{d}{=} X_{n,k}$. Thus P_k can be computed recursively by

$$\left\{\begin{aligned} P_0(z,y) &= y + \frac{z}{1-z}, \\ P_{k+1}(z,y) &= 1 + \int_0^z P_k^2(t,y)\,\mathrm{d}t \qquad (k\ge 0). \end{aligned}\right. \tag{2.1}$$

Explicit solutions (beyond the iterative integral forms) for this system of equations for all k seem intractable; we consider instead the moments of $X_{n,k}$ by expanding P_k as follows.

$$P_k(z,y) := \sum_{m\ge 0} \frac{M_{m,k}(z)}{m!}\,(y-1)^m,$$

so that $M_{m,k}(z) = \sum_n \mathbb{E}(X_{n,k}(X_{n,k}-1)\cdots(X_{n,k}-m+1))z^n$ and they satisfy, by (2.1),

$$M'_{m,k+1}(z) = \frac{2}{1-z}\, M_{m,k}(z) + \sum_{1\le j<m} \binom{m}{j} M_{j,k}(z) M_{m-j,k}(z) \tag{2.2}$$

for $k \ge 0$ and $m \ge 1$, with $M_{0,k}(z) = 1/(1-z)$ and $M_{m,k}(0) = 0$ $(k \ge 1)$.

More explicit representations for the $M_{m,k}$'s can be derived by considering the generating function

$$F_m(z,w) := \sum_{k\ge 0} M_{m,k}(z) w^k,$$

which satisfies, by (2.2), $F_m(0,w) = 0$ and

$$\frac{\partial}{\partial z} F_m(z,w) = \frac{2w}{1-z}\, F_m(z,w) + \sum_{1\le j<m} \binom{m}{j} \sum_{k\ge 0} M_{j,k}(z) M_{m-j,k}(z) w^{k+1}.$$

Solving this first-order differential equation yields

$$F_1(z,w) = (1-z)^{-2w}, \tag{2.3}$$

and for $m \ge 2$

$$F_m(z,w) = \sum_{1\le j<m} \binom{m}{j} (1-z)^{-2w} \int_0^z (1-t)^{2w} \sum_{k\ge 0} M_{j,k}(t) M_{m-j,k}(t) w^{k+1}\, \mathrm{d}t. \tag{2.4}$$

From (2.3), it follows that

$$M_{1,k}(z) = \frac{2^k}{k!} \log^k \frac{1}{1-z} \qquad (k \ge 0),$$

which implies (1.1), and then, by (2.4),

$$F_2(z,w) = 2w(1-z)^{-2w} \int_0^z (1-t)^{2w} I_0\left(4\sqrt{w} \log \frac{1}{1-t}\right) \mathrm{d}t, \tag{2.5}$$

where

$$I_0(z) = \sum_{k\ge 0} \frac{z^{2k}}{k!k!4^k}$$

is the modified Bessel function of order zero (see Abramowitz and Stegun (1965, section 9.6)).

Before going further, we derive an explicit formula for $\mathbb{E}(X_{n,k}(X_{n,k}-1))$.

Lemma 1. *The second factorial moments of $X_{n,k}$ can be computed by*

$$\mathbb{E}(X_{n,k}(X_{n,k}-1)) = \frac{2^k}{n!} \sum_{0\le j<k} \binom{2j}{j} 2^j \sum_{k+j-1\le m<n} s(n-1,m) \binom{m-2j-1}{k-j-1}. \tag{2.6}$$

Proof. First observe that

$$\begin{aligned}\int_0^1 (1-t)^{2w} I_0\left(4\sqrt{w}\log\frac{1}{1-t}\right)\mathrm{d}t &= \sum_{j\geq 0}\frac{4^j}{j!j!}\,w^j\int_0^1 y^{2w}\log^{2j}(1/y)\,\mathrm{d}y\\ &= \sum_{j\geq 0}\binom{2j}{j}\frac{(4w)^j}{(2w+1)^{2j+1}}\\ &= (4w^2-12w+1)^{-1/2}, \qquad (2.7)\end{aligned}$$

provided that $|\frac{16w}{(2w+1)^2}| < 1$. Assume for the moment that w lies in that region. Then, similarly as above,

$$\begin{aligned}&(1-z)^{-2w}\int_z^1 (1-t)^{2w} I_0\left(4\sqrt{w}\log\frac{1}{1-t}\right)\mathrm{d}t\\ &\quad= (1-z)\sum_{k\geq 0}\frac{(2k)!}{k!k!}(4w)^k\sum_{0\leq j\leq 2k}\frac{(-\log(1-z))^j}{j!}\cdot\frac{1}{(2w+1)^{2k+1-j}}\\ &\quad= \sum_{k\geq 0}\binom{2k}{k}(4w)^k\frac{1}{2\pi i}\oint_{|t|=c<|2w+1|}\frac{t^{-2k-1}(1-z)^{1-t}}{2w+1-t}\,\mathrm{d}t.\end{aligned}$$

But the residue of the integrand at $t = 2w+1$ equals $(2w+1)^{-2j-1}(1-z)^{-2w}$. It follows that

$$\begin{aligned}F_2(z,w) &= \frac{2w}{2\pi i}\oint_{|t|=c}\frac{(1-z)^{1-t}}{(t-2w-1)\sqrt{t^2-16w}}\,\mathrm{d}t \qquad (c > |2w+1|)\\ &= \frac{2w}{2\pi i}\oint_{|y|=c}\frac{(1-z)^{1-1/y}}{(1-(2w+1)y)\sqrt{1-16wy^2}}\,\mathrm{d}y \qquad (c < \varepsilon)\end{aligned}$$

for properly chosen integration contours. The restriction for w can now be dropped.

By Cauchy's integral representation

$$\mathbb{E}(X_{n,k}(X_{n,k}-1)) = \frac{2^k}{(2\pi i)^2}\oint\!\!\oint w^{-k}\frac{\binom{n-2+1/y}{n}}{(1-(w+1)y)\sqrt{1-8wy^2}}\,\mathrm{d}y\,\mathrm{d}w.$$

Thus we have

$$\mathbb{E}(X_{n,k}(X_{n,k}-1)) = 2^k\sum_{0\leq \ell<k}\binom{2\ell}{\ell}\frac{2^\ell}{2\pi i}\oint_{|z|=c>1}\frac{\binom{n+z-2}{n}}{z^{2\ell+1}(z-1)^{k-\ell}}\,\mathrm{d}z, \qquad (2.8)$$

from which (2.6) follows. □

3. Phase transitions and bimodality. *Notation.* For convenience, we use the symbol $[\![a,b]\!]$ to denote the interval $[a+K/\sqrt{\log n}, b-K/\sqrt{\log n}]$ for a sufficiently large K; The one-sided conventions $[a,b]\!]$ and $[\![a,b]$ stand for $[a, b-K/\sqrt{\log n}]$ and $[a+K/\sqrt{\log n}, b]$, respectively. The generic symbols K and ε always represent any large and small, respectively, numbers (independent of n and k) whose values may vary from one occurrence to another. Throughout this paper, $\alpha = \alpha_{n,k} = k/\log n$.

3.1. Asymptotics of $\mathbb{E}(X_{n,k})$. For completeness, we first state two known expansions for $\mathbb{E}(X_{n,k})$ that will be needed.

THEOREM 1. *Uniformly for* $1 \le k \le K \log n$,

$$\mathbb{E}(X_{n,k}) = \frac{(2\log n)^k}{nk!\Gamma(\alpha)} \left(1 + O\left((\log n)^{-1}\right)\right); \tag{3.1}$$

and uniformly for $k \to \infty$, $k \le K \log n$,

$$\mathbb{E}(X_{n,k}) \sim \frac{n^{\alpha - \alpha \log(\alpha/2) - 1}}{\sqrt{2\pi k}\,\Gamma(\alpha)} \sum_{j \ge 0} c_j k^{-j} \tag{3.2}$$

for some coefficients c_j.

Proof (sketch). The proof of both approximations starts from (1.1) and then uses the uniform approximation

$$\sum_k \mathbb{E}(X_{n,k}) w^k = \binom{n + 2w - 1}{n} = \frac{n^{2w-1}}{\Gamma(2w)} \left(1 + O(n^{-1})\right)$$

uniformly for $|w| \le K$ (by the singularity analysis of Flajolet and Odlyzko (1990)). Then

$$\mathbb{E}(X_{n,k}) = \frac{2^k}{2\pi i} \oint_{|w|=\alpha} w^{-k-1} \frac{n^{w-1}}{\Gamma(w)} \left(1 + O(n^{-1})\right) \mathrm{d}w,$$

and (3.1) follows by expanding $1/\Gamma(w)$ at $w = \alpha = k/\log n$, and by estimating the error terms properly; see Hwang (1995) for details. The proof for (3.2) uses the usual saddlepoint method and is similar. □

From (3.1), we see that the asymptotics of $\mathbb{E}(X_{n,k})/n$ is roughly dictated by a Poisson distribution with mean $2 \log n$. Also we can derive from (3.1) the local limit theorem for the depth and the upper bound $\alpha + \log n - \alpha' \log\log n + O(1)$ for the expected height, where $\alpha' = \alpha_+/(2\alpha_+ - 2)$; see Devroye (1987).

3.2. Asymptotics of $\mathbb{E}(X_{n,k}^2)$. For the second moment and the variance, the situation becomes completely different. We give our first approximations to $\mathbb{E}(X_{n,k}^2)$ by splitting the range $[0, K]$ into five nonoverlapping intervals.

Global silhouette. For simplicity of presentation, we drop the error terms in the following estimates and define two constants,

$$C_\pm := \frac{\sqrt{2} \pm 1}{2\sqrt{\pi\sqrt{2}}\,\Gamma(3 \pm 2\sqrt{2})}.$$

THEOREM 2. (I) *If* $\alpha \in [0, 3 - 2\sqrt{2}\rrbracket$, *then*

$$\mathbb{E}(X_{n,k}^2) \sim \mathbb{E}(X_{n,k}) \left(1 + \frac{\alpha}{\sqrt{\alpha^2 - 6\alpha + 1}}\right); \tag{3.3}$$

(II) *if* $\alpha \in \llbracket 3 - 2\sqrt{2}, 2 - \sqrt{2}\rrbracket$, *then*

$$\mathbb{E}(X_{n,k}^2) \sim C_- \frac{2^k n^{2-2\sqrt{2}} \left(3 - 2\sqrt{2}\right)^{-k}}{\sqrt{k - (3 - 2\sqrt{2}) \log n}}; \tag{3.4}$$

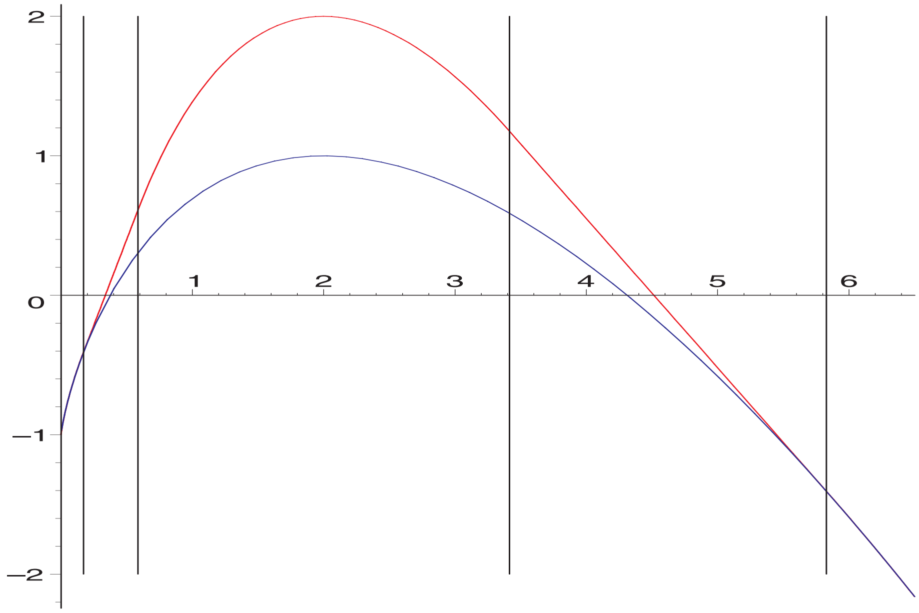


FIG. 3.1. *A plot of the limiting curve for* $\log \mathbb{E}(X_{n,k}^2)/\log n$ *(upper curve) and for* $\log \mathbb{E}(X_{n,k})/\log n$ *(lower curve) for* α *in each interval (horizontal coordinate). The intervals are also explicitly depicted by vertical lines.*

(III) *if* $\alpha \in [\![2-\sqrt{2}, 2+\sqrt{2}]\!]$, *then*

$$\mathbb{E}(X_{n,k}^2) \sim (\mathbb{E}X_{n,k})^2 \frac{\Gamma(\alpha)^2\alpha^2(2\alpha-1)}{\Gamma(2\alpha)(4\alpha-\alpha^2-2)}; \tag{3.5}$$

(IV) *if* $\alpha \in [\![2+\sqrt{2}, 3+2\sqrt{2}]\!]$, *then*

$$\mathbb{E}(X_{n,k}^2) \sim C_+ \frac{2^k n^{2+2\sqrt{2}} \left(3+2\sqrt{2}\right)^{-k}}{\sqrt{(3+2\sqrt{2})\log n - k}}; \tag{3.6}$$

(V) *finally, if* $\alpha \in [\![3+2\sqrt{2}, K]$, *then*

$$\mathbb{E}(X_{n,k}^2) \sim \mathbb{E}(X_{n,k}) \left(1 + \frac{\alpha}{\sqrt{\alpha^2-6\alpha+1}}\right). \tag{3.7}$$

A more transparent approximation is as follows; see Figure 3.1 for a plot.

COROLLARY 1 (phase transitions). *Let* $\bar\alpha := \lim_n k/\log n$. *The growth order of* $\mathbb{E}(X_{n,k}^2)$ *satisfies*

$$\frac{\log \mathbb{E}(X_{n,k}^2)}{\log n} \to \begin{cases} \bar\alpha - \bar\alpha\log(\bar\alpha/2) - 1 & \text{if } \bar\alpha \in [0, 3-2\sqrt{2}], \\ 2-2\sqrt{2}-2\bar\alpha\log(1-2^{-1/2}) & \text{if } \bar\alpha \in [3-2\sqrt{2}, 2-\sqrt{2}], \\ 2(\bar\alpha - \bar\alpha\log(\bar\alpha/2) - 1) & \text{if } \bar\alpha \in [2-\sqrt{2}, 2+\sqrt{2}], \\ 2+2\sqrt{2}-2\bar\alpha\log(1+2^{-1/2}) & \text{if } \bar\alpha \in [2+\sqrt{2}, 3+2\sqrt{2}], \\ \bar\alpha - \bar\alpha\log(\bar\alpha/2) - 1 & \text{if } \bar\alpha \in [3+2\sqrt{2}, K]. \end{cases}$$

By continuity, the (almost) open boundaries $[\![$ and $]\!]$ in all cases are replaced by the closed ones [and], respectively, as will become clear.

Transitional behaviors. These quick (and rough) estimates leave open the asymptotics of the second moment in the transitional ranges $k=(3\pm 2\sqrt{2})\log n+O(\sqrt{\log n})$ and $k=(2\pm\sqrt{2})\log n+O(\sqrt{\log n})$, which will be handled by more uniform asymptotic tools.

Let $D_{-\nu}(x)$ denote the parabolic cylinder function (see Abramowitz and Stegun (1965, Ch. 19)), which can be defined by

$$D_{-\nu}(x)=\frac{e^{-x^2/4}}{\Gamma(\nu)}\int_0^\infty u^{\nu-1}e^{-xu-u^2/2}\,\mathrm{d}u \qquad (\nu>0), \tag{3.8}$$

and let $\Phi(x)$ denote the standard normal distribution function. Note that $\Phi(x)$ is itself a special case of the parabolic cylinder functions

$$D_{-1}(x)=\sqrt{2\pi}\,e^{x^2/4}\Phi(-x).$$

Theorem 3. *All asymptotic estimates below hold uniformly for $t=o((\log n)^{1/6})$.* (i) *If $\alpha=3-2\sqrt{2}+(\sqrt{2}-1)t/\sqrt{\log n}$, then*

$$\mathbb{E}(X_{n,k}^2)\sim 2^{-1/2}C_-e^{t^2/4}D_{-1/2}(-t)k^{-1/4}\,n^{2-2\sqrt{2}-2\alpha\log(1-1/\sqrt{2})}; \tag{3.9}$$

(ii) *if $\alpha=2-\sqrt{2}+\sqrt{1-2^{-1/2}}t/\sqrt{\log n}$, then*

$$\mathbb{E}(X_{n,k}^2)\sim 2^{1/4}C_-\Phi(-t)k^{-1/2}n^{2-2\sqrt{2}-2\alpha\log(1-1/\sqrt{2})}; \tag{3.10}$$

(iii) *if $\alpha=2+\sqrt{2}+\sqrt{1+2^{-1/2}}t/\sqrt{\log n}$, then*

$$\mathbb{E}(X_{n,k}^2)\sim 2^{1/4}C_+\Phi(t)k^{-1/2}n^{2+2\sqrt{2}-2\alpha\log(1+1/\sqrt{2})};$$

(iv) *finally, if $\alpha=3+2\sqrt{2}+(\sqrt{2}+1)t/\sqrt{\log n}$, then*

$$\mathbb{E}(X_{n,k}^2)\sim 2^{-1/2}C_+e^{t^2/4}D_{-1/2}(t)k^{-1/4}\,n^{2+2\sqrt{2}-2\alpha\log(1+1/\sqrt{2})}.$$

In all cases, the dropped error terms are of the form

$$1+O\left(\frac{1+|t|^3}{\sqrt{\log n}}\right).$$

These estimates complete the gaps left open in Theorem 2; furthermore, one can easily check that in the overlapping ranges ($K\le|t|=o((\log n)^{1/6})$) the approximations in both theorems coincide by the following asymptotic estimates (see Abramowitz and Stegun (1965, section 19.7)):

$$\begin{cases} D_{-\nu}(x)\sim x^{-\nu}e^{-x^2/4} & (x\to\infty),\\ D_{-\nu}(-x)\sim \dfrac{\sqrt{2\pi}}{\Gamma(\nu)}x^{\nu-1}e^{x^2/4} & (x\to\infty). \end{cases} \tag{3.11}$$

Bimodality. Everything up to now is only unimodal. Bimodality of the variance appears in the middle range $\alpha\in[\![2-\sqrt{2},2+\sqrt{2}]\!]$.

First, from Theorem 2, we readily obtain the following estimate.

Corollary 2. *The variance of $X_{n,k}$ satisfies*

$$\mathbb{V}(X_{n,k})\sim\varphi(\alpha)(\mathbb{E}X_{n,k})^2$$

for $\alpha \in [\![2-\sqrt{2}, 2+\sqrt{2}]\!]$*, where* φ *is defined in* (1.3)*, and* $\mathbb{V}(X_{n,k}) \sim \mathbb{E}(X_{n,k}^2)$ *for all other ranges.*

Observe that

$$\varphi(1) = \varphi(2) = \varphi'(1) = \varphi'(2) = 0; \tag{3.12}$$

thus the estimate (1.3) is insufficient for an asymptotic equivalent for the variance in the central range $k = (2+o(1))\log n$ and in the somewhat unexpected range $k = (1+o(1))\log n$. We need stronger approximations.

Theorem 4. *If* $\alpha \in [\![2-\sqrt{2}, 2+\sqrt{2}]\!]$*, then*

$$\mathbb{V}(X_{n,k}) \sim n^{2(\alpha-\alpha\log(\alpha/2)-1)} \sum_{j\geq 1} \frac{\upsilon_j(\alpha)}{(\log n)^j} \tag{3.13}$$

for some coefficients $\upsilon_j(\alpha)$*; see* (5.1) *and* (5.3) *below.*

In particular, $\upsilon_1(\alpha) = \varphi(\alpha)/(2\pi\alpha\Gamma(\alpha)^2)$ also satisfies property (3.12), and $\upsilon_2(\alpha)$ satisfies $\upsilon_2(1) = \upsilon_2(2) = 0$.

Corollary 3. *If* $\alpha = 2 + t/\log n$*, where* $t = o(\log n)$*, then*

$$\mathbb{V}(X_{n,k}) = \frac{p_1(t)}{720\pi} \cdot \frac{n^{2(\alpha-\alpha\log(\alpha/2)-1)}}{(\log n)^3}\left(1 + O\left(\frac{1+|t|}{\log n}\right)\right)$$

uniformly in t*, where* $p_1(t)$ *is a quadratic polynomial defined by*

$$\begin{aligned} p_1(t) := {} & 15(21-2\pi^2)t^2 - 30\left(4\pi^2(1-\gamma) + 24\zeta(3) + 42\gamma - 69\right)t \\ & - 2\pi^4 - 30\left(4\gamma^2 - 8\gamma + 11\right)\pi^2 + 180\left(7\gamma^2 - 23\gamma + 29\right) - 1440\zeta(3)(1-\gamma), \end{aligned}$$

where γ *denotes Euler's constant and* $\zeta(3) := \sum_{j\geq 1} j^{-3}$.

The reason of writing the corollary in its form is that the variation of the order of $\mathbb{V}(X_{n,k})$ when $k = (2+o(1))\log n$ becomes more transparent. Thus, if $\alpha = 2+t/\log n$, where $t = o((\log n)^{3/2})$, then

$$\mathbb{V}(X_{n,k}) \sim \frac{p_1(t)}{720\pi} \cdot \frac{n^2}{(\log n)^3} \exp\left(-\frac{t^2}{2\log n}\right)$$

uniformly in t. From this we can derive approximations to the scale of the two "humps" and the valley shown in Figure 1.1.

Corollary 4. *The largest value of* $\mathbb{V}(X_{n,k})$ *is asymptotically achieved at* $k = \left\lfloor 2\log n \pm \sqrt{2\log n}\right\rfloor$*, and*

$$\max_{k\geq 0} \mathbb{V}(X_{n,k}) \sim \frac{21-2\pi^2}{48\pi e} \cdot \frac{n^2}{(\log n)^2};$$

on the other hand,

$$\min_{|k-2\log n| = O(\sqrt{\log n})} \mathbb{V}(X_{n,k}) \geq (C + o(1))\frac{n^2}{(\log n)^3}, \tag{3.14}$$

where

$$C = \frac{4\pi^6 + 378\pi^4 - 9090\,\pi^2 - 38205 - 8640\zeta(3)^2 + 19440\zeta(3) - 38205}{720\pi(21-2\pi^2)}.$$

The smallest value of $\mathbb{V}(X_{n,k})$*, for* $k = 2\log n + O(\sqrt{\log n})$*, is asymptotically achieved only for the subsequence of* n *for which* $\{2\log n\} \to 1 - t_0$*, where*

$$t_0 = -2(1-\gamma) + \frac{3(8\zeta(3)-9)}{21-2\pi^2} \approx 0.62126\ldots$$

satisfies $p_1'(t_0) = 0$.

Thus *the variance can vary from* $n^2/(\log n)^2$ *to* $n^2/(\log n)^3$ *when* $k = 2\log n + O(\sqrt{\log n})$*, and these are precisely the orders of the peak and the valley, respectively, as shown in Figure* 1.1.

Our analysis here says that the two peaks are asymptotically of the same order, although Figure 1.1 may lead one to guess that the left peak is higher. We will see that this is indeed true by further examining the sign of the next order term; see section 5 for more details.

A "false valley."

Corollary 5. *If* $\alpha = 1 + t/\log n$*, where* $t = o((\log n)^{2/3})$*, then*

$$\mathbb{V}(X_{n,k}) \sim \frac{4^t \varpi(t)}{720\pi} \cdot \frac{n^{2\log 2}}{(\log n)^3}\, e^{-t^2/\log n},$$

uniformly in t*, where* $\varpi(t)$ *is defined by*

$$\begin{aligned}\varpi(t) := {}& 60(12-\pi^2)t^2 + 120\left(\pi^2\gamma - 12\gamma - 6\zeta(3) + 12\right)t \\ & - \pi^4 - 60\left(\gamma^2+2\right)\pi^2 + 720\left(\gamma^2 - 2\gamma + \zeta(3)\gamma + 3\right).\end{aligned}$$

One sees that although the order of the variance can reach that of $\mathbb{E}(X_{n,k}^2)/(\log n)^2$ (when $k = \log n + O(1)$) as in the case $k = 2\log n + O(1)$, there is no new "valley" generated when $k = \log n + O(\sqrt{\log n})$ since the logarithmically smaller terms are "smoothed out" by an exponentially larger factor 4^t.

4. Phase transitions: Proof of Theorem 2. For more methodological interest and shedding more light on how the different ranges arise, we give in this section two proofs of Theorem 2. The first relies essentially on the exact expression (2.6), which has some elementary flavor, although the main estimate needed relies on saddlepoint method. The error estimates obtained by this approach are, however, insufficient for describing the bimodal behavior of the variance. The second proof uses (2.5) and is analytic in nature; it is needed to complete the transitional behaviors of Theorem 3 and can be easily extended to derive asymptotic expansions. In particular, it gives the precise description of the valley and the required error bounds in all cases.

4.1. A direct approach. We give in this section the sketch of an approach to proving Theorem 2 using (2.6). The basic idea is first to find a good uniform estimate for the sum

$$S_{n,k,j} := \sum_{k+j-1\le m<n} \frac{s(n-1,m)}{n!}\binom{m-2j-1}{k-j-2} \qquad (0 \le j < k);$$

then we evaluate the sum

$$\mathbb{E}(X_{n,k}(X_{n,k}-1)) = 2^k \sum_{0\le j<k} \binom{2j}{j} 2^j S_{n,k,j},$$

by different means according to the range of α.

In this subsection, we always write $\alpha = k/\log n$ and $\lambda = j/\log n$.

LEMMA 2. *Define*

$$f(z) = f(\alpha, \lambda; z) := z - 2\lambda \log z - (\alpha - \lambda)\log(z-1),$$

and

$$z_0 = z_0(\alpha, \lambda) := \frac{\alpha+\lambda+1}{2} + \sqrt{\left(\frac{\alpha+\lambda+1}{2}\right)^2 - 2\lambda}.$$

If $1+\varepsilon \le z_0 \le K$, *then*

$$S_{n,k,j} \sim \frac{n^{f(z_0)-2}}{z_0\Gamma(z_0-1)\sqrt{2\pi f''(z_0)\log n}},$$

uniformly in k *and* j.

Proof. We start from the integral representation (see (2.8))

$$\begin{aligned} S_{n,k,j} &= \frac{1}{2\pi i}\oint_{|z|=z_0} \frac{\binom{n+z-2}{n}}{z^{2j+1}(z-1)^{k-j}}\,\mathrm{d}z \\ &= \frac{1}{2\pi i}\oint_{|z|=z_0} \frac{n^{z-2}}{\Gamma(z-1)z^{2j+1}(z-1)^{k-j}}\left(1+O(n^{-1})\right)\,\mathrm{d}z \end{aligned}$$

by singularity analysis. Observe that z_0 is the saddlepoint for which $f'(z_0)=0$, and that the second derivative of f,

$$f''(z) = \frac{2\lambda}{z^2} + \frac{\alpha-\lambda}{(z-1)^2},$$

remains strictly positive in the range of interest. The required result follows from applying the saddlepoint method to the integral

$$\frac{1}{2\pi i}\oint_{|z|=z_0} \frac{e^{\log n f(z)}}{z\Gamma(z-1)}\,\mathrm{d}z. \qquad \square$$

Middle range. Consider first case (III): $\alpha \in [\![2-\sqrt{2}, 2+\sqrt{2}]\!]$. In this case terms with large j's are dominant. Thus, we set $r := k-j \ge 1$. By applying Lemma 2 with $\lambda = \alpha - r/\log n$,

$$z_0 = 2\alpha - \frac{2r(\alpha-1)}{(2\alpha-1)\log n} + O\left((\log n)^{-2}\right),$$

and

$$f(\alpha,\lambda;z_0) = 2\alpha + r\frac{2\log(2\alpha) - \log(2\alpha-1)}{\log n} + O\left((\log n)^{-2}\right),$$

we get

$$S_{n,k,j} \sim \left(\frac{4\alpha^2}{2\alpha-1}\right)^r \frac{n^{f(\alpha,\alpha;z_0)-2}}{z_0\Gamma(z_0-1)\sqrt{2\pi\log n/(2\alpha)}};$$

also

$$2^j\binom{2j}{j} \sim \frac{8^{k-r}}{\sqrt{(k-r)\pi}}.$$

These estimates lead to

$$\begin{aligned}\mathbb{E}(X_{n,k}(X_{n,k}-1)) &\sim \frac{16^k}{\sqrt{k\pi}}\sum_{r\ge 1}\left(\frac{4\alpha^2}{8(2\alpha-1)}\right)^r \frac{n^{f(\alpha,\alpha;2\alpha)-2}}{z_0\Gamma(z_0-1)\sqrt{2\pi\log n/(2\alpha)}}\\ &= \frac{\alpha^2}{\Gamma(2\alpha-1)(4\alpha-\alpha^2-2)}\cdot\frac{4^k e^{2k}(\log n)^{2k}}{2\pi k n^2 k^{2k}}\\ &\sim \frac{\alpha^2(2\alpha-1)}{\Gamma(2\alpha)(4\alpha-\alpha^2-2)}\cdot\left(\frac{(2\log n)^k}{n\,k!}\right)^2.\end{aligned}$$

Intermediate ranges. For case (IV), $\alpha \in [\![2+\sqrt{2}, 3+2\sqrt{2}]\!]$, no terms are asymptotically negligible; we then sum all terms up and obtain

$$\mathbb{E}(X_{n,k}(X_{n,k}-1)) \sim \frac{2^k}{\sqrt{2\pi}n^2\log n}\sum_{1\le j<k}\frac{n^{F(\lambda)}}{\sqrt{\lambda}z_0\Gamma(z_0-1)\sqrt{f''(z_0)}},$$

where $F(\lambda) := \lambda\log 8 + f(\alpha,\lambda;z_0(\alpha,\lambda))$. Since $f'(\alpha,\lambda,z_0(\alpha,\lambda)) = 0$, we get $F'(\lambda) = \log 8 - 2\log z + \log(z-1)$; and, consequently, $F'(\lambda_0) = 0$ for $z_0(\alpha,\lambda_0) = 2(2+\sqrt{2})$, which implies that $\lambda_0 = \sqrt{2}(3+2\sqrt{2}-\alpha)$. It follows that $F(\lambda_0) = 4+2\sqrt{2}-\alpha\log(3+2\sqrt{2})$, $F''(\lambda_0) = -\sqrt{2}/(5+4\sqrt{2}+\alpha)$, and

$$f''(\alpha,\lambda_0;2(2+\sqrt{2})) = \frac{1}{4}(17\sqrt{2}-24)(5+4\sqrt{2}+\alpha);$$

we obtain, by standard application of the saddlepoint method,

$$\begin{aligned}\mathbb{E}(X_{n,k}(X_{n,k}-1)) &\sim \frac{2^k}{\sqrt{2}z_0\Gamma(z_0-1)\pi n^2\log n}\sqrt{\frac{2\pi\log n}{-\lambda_0 F''(\lambda_0)f''(z_0)}}\,n^{F(\lambda_0)}\\ &= \frac{2^k n^{2+2\sqrt{2}}(3+2\sqrt{2})^{-k}}{\sqrt{2\pi\log n}(2-\sqrt{2})\Gamma(3+2\sqrt{2})\sqrt{\sqrt{2}(3+2\sqrt{2}-\alpha)}}.\end{aligned}$$

This proves (3.6). The proof for case (II) is similar.

Extremal ranges. Case (V): $\alpha \in [\![3+2\sqrt{2}, K]$. In this case, the terms with small j are dominant. For every finite $j \ge 0$, we have ($z_0 = \alpha+1$)

$$\begin{aligned}S_{n,k,j} &\sim \frac{1}{2\pi i}\oint_{|z|=z_0}\frac{n^{z-2}}{\Gamma(z-1)z^{2k+1}}\left(\frac{z-1}{z^2}\right)^j \mathrm{d}z\\ &\sim \left(\frac{\alpha}{(\alpha+1)^2}\right)^j\frac{n^{\alpha-1-\alpha\log\alpha}}{(\alpha+1)\Gamma(\alpha)\sqrt{2\pi\log n/\alpha}}\\ &\sim \left(\frac{\alpha}{(\alpha+1)^2}\right)^j\frac{\alpha(\log n)^k}{(\alpha+1)\Gamma(\alpha)nk!}.\end{aligned}$$

Consequently,

$$\begin{aligned}\mathbb{E}(X_{n,k}(X_{n,k}-1)) &\sim \frac{(2\log n)^k\alpha}{n\,k!\,(\alpha+1)\Gamma(\alpha)}\sum_{j\ge 0}\binom{2j}{j}\left(\frac{2\alpha}{(\alpha+1)^2}\right)^j\\ &= \frac{\alpha(2\log n)^k}{(\alpha+1)\Gamma(\alpha)nk!}\left(1-\frac{8\alpha}{(\alpha+1)^2}\right)^{-1/2}\\ &= \frac{\alpha(2\log n)^k}{\Gamma(\alpha)\sqrt{\alpha^2-6\alpha+1}nk!}.\end{aligned}$$

Case (I) is similar.

4.2. An analytic approach. This approach relies on (2.5), and the convergence or divergence of the integral

$$\int_0^1 (1-t)^{2w} I_0\left(4\sqrt{w}\log\frac{1}{1-t}\right)\,\mathrm{d}t \tag{4.1}$$

plays a crucial rôle in determining the different ranges.

We first give the main idea of this approach using mostly heuristic reasoning; the technical justification and detailed estimates of the error terms will be provided later.

A sketch of proof. We need the asymptotics of the modified Bessel function (see Abramowitz and Stegun (1965, section 9.6)):

$$I_0(z) = \frac{e^z}{\sqrt{2\pi z}}\left(1+O(|z|^{-1})\right), \tag{4.2}$$

the O-term being uniform for $|z|\to\infty$ in the region $-\pi/2 < \arg(z) \le \pi/2$.

Small or large α. First, if the integral (4.1) is convergent, then (see (2.7))

$$\begin{aligned}F_2(z,w) &\sim 2w(1-z)^{-2w}\int_0^1 (1-t)^{2w} I_0\left(4\sqrt{w}\log\frac{1}{1-t}\right)\,\mathrm{d}t\\ &= \frac{2w}{\sqrt{4w^2-12w+1}}(1-z)^{-2w}; \qquad (4.3)\end{aligned}$$

so we expect that (by singularity analysis and then by the saddlepoint method)

$$\begin{aligned}\mathbb{E}(X_{n,k}(X_{n,k}-1)) &= [w^kz^n]F_2(z,w)\\ &\sim [w^k]\frac{2wn^{2w-1}}{\sqrt{4w^2-12w+1}\,\Gamma(2w)}\\ &\sim \frac{\alpha}{\sqrt{\alpha^2-6\alpha+1}\,\Gamma(\alpha)}\cdot\frac{(2\log n)^k}{nk!},\end{aligned}$$

where $\alpha > 0$ has to satisfy $\alpha^2-6\alpha+1>0$. This gives rise to the first two ranges $\alpha\in[0,3-2\sqrt{2})$ and $\alpha\in(3+2\sqrt{2},K]$, and the estimates (3.3) and (3.7).

Middle range. On the other hand, if the integral (4.1) diverges, then by (4.2)

$$\begin{aligned}F_2(z,w) &\sim 2w(1-z)^{-2w}\int_0^z (1-t)^{2w} I_0\left(4\sqrt{w}\log\frac{1}{1-t}\right)\,\mathrm{d}t\\ &\sim \frac{2w(1-z)^{-2w}}{\sqrt{8\pi\sqrt{w}}}\int_0^z\left(\log\frac{1}{1-t}\right)^{-1/2}(1-t)^{2w-4\sqrt{w}}\,\mathrm{d}t\\ &\sim \frac{w}{\sqrt{2\pi\sqrt{w}}(4\sqrt{w}-2w-1)}\left(\log\frac{1}{1-z}\right)^{-1/2}(1-z)^{-4\sqrt{w}+1}. \qquad (4.4)\end{aligned}$$

Thus we expect that (again by singularity analysis and then by the saddlepoint method)

$$\begin{aligned}\mathbb{E}(X_{n,k}(X_{n,k}-1)) &\sim [w^k]\frac{wn^{4\sqrt{w}-2}(\log n)^{-1/2}}{\sqrt{2\pi\sqrt{w}}(4\sqrt{w}-2w-1)\Gamma(4\sqrt{w}-1)}\\ &\sim \frac{\alpha^2}{(4\alpha-\alpha^2-2)\Gamma(2\alpha-1)}\cdot\frac{(2\log n)^{2k}}{n^2(k!)^2},\end{aligned}$$

which yields the second pairs of transitional points since

$$4\alpha-\alpha^2-2>0 \text{ iff } \alpha\in(2-\sqrt{2},2+\sqrt{2}).$$

Intermediate ranges. Observe that the error term in (4.3) is of the form (by (4.2))

$$\begin{aligned}(1-z)^{-2\Re(w)}&\int_z^1(1-t)^{2w}I_0\left(4\sqrt{w}\log\frac{1}{1-t}\right)\mathrm{d}t\\ &=O\left(\frac{(1-z)^{-4\sqrt{w}+1}}{|4\sqrt{w}-2w-1|}\left(\log\frac{1}{1-z}\right)^{-1/2}\right)\end{aligned}$$

(see also (4.4)), whose contribution to $\mathbb{E}(X_{n,k}(X_{n,k}-1))$ is roughly of the order

$$[w^k]\frac{n^{4\sqrt{w}-2}}{4\sqrt{w}-2w-1}(\log n)^{-1/2}=O\left(\frac{(2\log n)^{2k}}{n^2k!^2}\right),$$

essentially of the same order as $(\mathbb{E}(X_{n,k}))^2$.

Thus we can use the estimate (4.3) when k lies in the intervals of cases (II) and (IV); but instead of applying the saddlepoint method as in cases (I) and (V), we use again singularity analysis since the singularities at $w=\frac{3}{2}\pm\sqrt{2}$ in (4.3) is dominating.

Consider case (II). Let $\beta:=3/2-\sqrt{2}$. We have, by (4.3),

$$\begin{aligned}\mathbb{E}(X_{n,k}(X_{n,k}-1)) &\sim [w^k]\frac{2w}{\sqrt{4w^2-12w+1}}\cdot\frac{n^{2w-1}}{\Gamma(2w)}\\ &\sim \frac{2\beta n^{2\beta-1}}{\sqrt{8\sqrt{2}\beta}\,\Gamma(2\beta)}[w^k]\frac{n^{2(w-\beta)}}{\sqrt{1-w/\beta}}\\ &\sim \frac{n^{2-2\sqrt{2}}}{\sqrt{2\pi\sqrt{2}}\,\Gamma(3-2\sqrt{2})}\left(\frac{3}{2}-\sqrt{2}\right)^{-k+1/2}(k-2\beta\log n)^{-1/2},\end{aligned}$$

which, in view of (3.1), implies (3.4).

Case (IV) is similar.

4.3. Technical justification and error estimates. We start from deriving a different integral representation for F_2 suitable for all ranges.

LEMMA 3.

$$F_2(z,w)=\frac{2w}{\pi}\int_{-1}^1\frac{(1-z)^{-2w}-(1-z)^{-4\sqrt{w}v+1}}{\sqrt{1-v^2}(2w+1-4\sqrt{w}v)}\,\mathrm{d}v. \tag{4.5}$$

Note that this representation is well-defined for all w (including at the zeros of the factors in the denominator).

Proof. By the integral representation for $I_0(z)$ (see Abramowitz and Stegun (1965, p. 376))

$$I_0(z) = \frac{1}{\pi}\int_0^\pi e^{z\cos t}\,\mathrm{d}t,$$

and by (2.5), we have

$$\begin{aligned} F_2(z,w) &= \frac{2w}{\pi}(1-z)^{-2w}\int_0^z (1-t)^{2w}\int_0^\pi (1-t)^{-4\sqrt{w}\cos y}\,\mathrm{d}y\,\mathrm{d}t \\ &= \frac{2w}{\pi}(1-z)^{-2w}\int_{-1}^1 \frac{1}{\sqrt{1-v^2}}\int_0^z (1-t)^{2w-4\sqrt{w}v}\,\mathrm{d}t\,\mathrm{d}v, \end{aligned}$$

which yields (4.5). □

Note that when $w \notin [3/2-\sqrt{2}, 3/2+\sqrt{2}]$ we can split the integral (4.5) and obtain

$$\begin{aligned} F_2(z,w) = {}& \frac{2w}{\pi}(1-z)^{-2w}\int_{-1}^1 \frac{\mathrm{d}v}{\sqrt{1-v^2}(2w+1-4\sqrt{w}v)} \\ &+ \frac{2w}{\pi}\int_{-1}^1 \frac{(1-z)^{-4\sqrt{w}v+1}}{\sqrt{1-v^2}(4\sqrt{w}v-2w-1)}\,\mathrm{d}v \\ ={}& \frac{2w(1-z)^{-2w}}{\sqrt{4w^2-12w+1}} + \frac{2w}{\pi}\int_{-1}^1 \frac{(1-z)^{-4\sqrt{w}v+1}}{\sqrt{1-v^2}(4\sqrt{w}v-2w-1)}\,\mathrm{d}v. \end{aligned}$$

Roughly, when k lies in the middle range, the main contribution comes from the second integral, which becomes asymptotically negligible for k outside that range.

Proposition 1. *Uniformly for $\alpha \le K$,*

$$[w^kz^n]F_2(z,w) = [w^k]\frac{2w}{\pi}\int_{-1}^1 \frac{1}{\sqrt{1-v^2}(2w+1-4\sqrt{w}v)}\left(\frac{n^{2w-1}}{\Gamma(2w)} - \frac{n^{4\sqrt{w}v-2}}{\Gamma(4\sqrt{w}v-1)}\right)\mathrm{d}v + T_1, \tag{4.6}$$

where

$$T_1 = O\left(\frac{(2\log n)^k}{n^2k!}\sqrt{k}\log n + \frac{(2\log n)^{2k}}{n^3k!^2}\,k\log n\right).$$

Proof. By singularity analysis (see Flajolet and Odlyzko (1990)), we have

$$[z^n](1-z)^{-\omega} = \frac{n^{\omega-1}}{\Gamma(\omega)}\left(1+\frac{\omega(\omega-1)}{2n} + O\left(n^{-2}\right)\right)$$

uniformly for $|\omega| \le K$. Note that if $4\sqrt{w}v \sim 2w+1$, then

$$[z^n]\frac{(1-z)^{-2w}-(1-z)^{-4\sqrt{w}v+1}}{2w+1-4\sqrt{w}v} = O\left(n^{2\Re(w)-1}\log n\right).$$

Thus

$$[z^n]F_2(z,w) = \frac{2w}{\pi}\int_{-1}^1 \frac{1}{\sqrt{1-v^2}(2w+1-4\sqrt{w}v)}\left(\frac{n^{2w-1}}{\Gamma(2w)} - \frac{n^{4\sqrt{w}v-2}}{\Gamma(4\sqrt{w}v-1)}\right)\mathrm{d}v + T_2,$$

where

$$T_2 = O\left(n^{2\Re(w)-2}\log n + n^{4\Re(\sqrt{w})-3}\log n\right).$$

Now by Cauchy's integral formula

$$\begin{aligned}[w^k]T_2 &= O\left(r_1^{-k}n^{2r_1-2}\log n + r_2^{-k}n^{4\sqrt{r}-3}\log n\right)\\ &= O\left(n^{\alpha-\alpha\log(\alpha/2)-2}\log n + n^{2(\alpha-\alpha\log(\alpha/2)-1)-2}\log n\right)\end{aligned}$$

by taking $r_1 = \alpha/2$ and $r_2 = (\alpha/2)^2$. Thus (4.6) follows. □

Cases (I) *and* (V). Consider first case (I). With the uniform estimate (4.6) at hand, we obtain the leading term in (3.3) by expanding the factor

$$H(w) := \frac{2w}{\sqrt{4w^2-12w+1}\,\Gamma(2w)}$$

at $w = \alpha/2$ and then use the saddlepoint method; see Hwang (1995) for similar details. It remains to show, again by (4.6), that the integral

$$T_3 := \frac{2}{\pi}\cdot\frac{1}{2\pi i}\oint_{|w|=r} w^{-k}\int_{-1}^{1}\frac{n^{4\sqrt{w}v-2}}{\sqrt{1-v^2}(2w+1-4\sqrt{w}v)\Gamma(4\sqrt{w}v-1)}\,\mathrm{d}v\,\mathrm{d}w,$$

where $r := (\alpha/2)^2$, satisfies

$$T_3 = O\left(\frac{(2\log n)^{2k}}{n^2 k!^2|\alpha^2-4\alpha+2|}\right) \tag{4.7}$$

uniformly for $\alpha \in [0, 3-2\sqrt{2}]$. Indeed, we prove that this estimate holds uniformly for $|\alpha-(2\pm\sqrt{2})| \geq K/\sqrt{\log n}$.

By the elementary inequality $1-\cos t \geq 2t^2/\pi^2$ for $|t| \leq \pi$, we have

$$n^{4\Re(\sqrt{w})v} = n^{4\sqrt{r}v\cos(t/2)} \leq n^{2\alpha v-\alpha vt^2/\pi^2} = e^{2kv-kvt^2/\pi^2} \qquad (|t| \leq \pi),$$

so that the major contribution to T_3 comes from the ranges

$$1-\varepsilon \leq v \leq 1 \quad \text{and} \quad \{w = re^{it} : |t| \leq \varepsilon\},$$

the integrals over the remaining ranges being bounded above by

$$O\left(n^{2(\alpha-\alpha\log(\alpha/2)-1)-\varepsilon}\right).$$

Thus when $|2r+1-4\sqrt{r}| = |\alpha^2-4\alpha+2|/2 \geq \varepsilon$

$$\begin{aligned}T_3 &= O\left(\frac{(\alpha/2)^{-2k}n^{-2}}{|\alpha^2-4\alpha+2|}\int_{|t|\leq\varepsilon} e^{2k-kt^2/\pi^2}\int_0^{(\log n)^{-3/5}} u^{-1/2}e^{-2ku}\,\mathrm{d}u\,\mathrm{d}t\right)\\ &= O\left(\frac{(\alpha/2)^{-2k}e^{2k}n^{-2}}{|\alpha^2-4\alpha+2|k}\right),\end{aligned}$$

from which we obtain (4.7). By examining further the second order terms (see (5.2) below), we can take $\varepsilon = K/\sqrt{\log n}$. This proves (3.3).

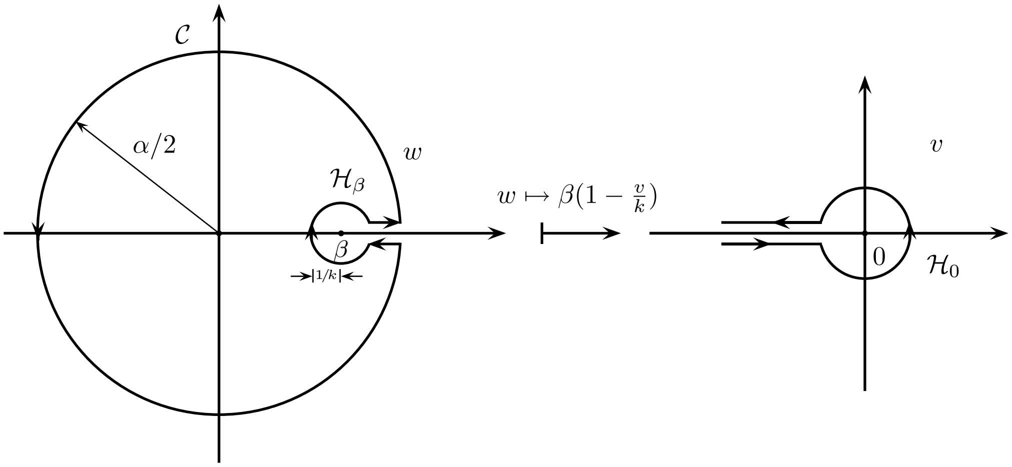


Fig. 4.1. *The Hankel type contours used for proving the estimate in case (*II*).*

The estimate (3.7) is similar.

Cases (II) *and* (IV). Consider first case (II). Since there is a singularity at $w = \beta := 3/2 - \sqrt{2}$, we apply again singularity analysis to the integral

$$\begin{aligned} T_4 &:= \frac{1}{2\pi i}\oint_{|w|=r} H(w) w^{-k-1} n^{2w-1}\,\mathrm{d}w \\ &= \frac{1}{2\pi i}\oint_{|w|=r} \frac{h(w)}{\sqrt{\beta - w}}\, w^{-k} n^{2w-1}\,\mathrm{d}w, \end{aligned} \tag{4.8}$$

where $0 < r < \beta$ and

$$h(w) := \frac{2}{\Gamma(2w)}\sqrt{\frac{\beta - w}{4w^2 - 12w + 1}},$$

the principal branch being taken so that $h(w) > 0$ for $0 < w < \beta$. The integration circle is then deformed into the one shown in Figure 4.1, where the smaller circle (left) is described by $|w - \beta| = 1/k$.

The contribution to T_4 from the outer circle $\mathcal{C}$ is easily seen to be of order

$$O\left(\frac{n^{\alpha - \alpha\log(\alpha/1) - 1}}{\sqrt{(2\beta - \alpha)\log n}}\right).$$

For the integral along the contour $\mathcal{H}_\beta$, we make the change of variables $w \mapsto \beta(1 - v/k)$, so that $\mathcal{H}_\beta$ is transformed into $\mathcal{H}_0$ (also shown in Figure 4.1). Then

$$\begin{aligned} T_4 &= k^{-1/2}\beta^{-k+1/2} n^{2\beta-1} \cdot \frac{1}{2\pi i}\int_{\mathcal{H}_0} h\left(\beta\left(1 - v/k\right)\right) v^{-1/2} e^{v(1-2\beta/\alpha)}\left(1 + O(|v|^2 k^{-1})\right)\mathrm{d}v \\ &\qquad + O\left(\frac{n^{\alpha - \alpha\log(\alpha/2) - 1}}{\sqrt{(2\beta - \alpha)\log n}}\right) \\ &= \frac{h(\beta)}{\sqrt{\pi(1 - 2\beta/\alpha)}} k^{-1/2}\beta^{-k+1/2} n^{2\beta-1}\left(1 + O\left(\frac{1}{(\alpha - 2\beta)^2 k}\right)\right), \end{aligned}$$

from which (3.4) follows since $\beta^{1/2} = 1 - 2^{-1/2}$ and $h(\beta) = 2^{-3/4}/\Gamma(2\beta)$; see Flajolet and Odlyzko (1990) for similar details. The error term yields exactly the left boundary $\alpha \ge (3-2\sqrt{2})\log n + K\sqrt{\log n}$; the right boundary $(2-\sqrt{2})\log n - K\sqrt{\log n}$ comes from (4.6).

For the estimate (3.6), the proof is similar. Note that since $H(w)$ has a singularity at $w = \beta$, we have to start from (4.6) and then proceed similarly.

Middle range. We use again (4.6). The same observation that the major contribution comes from $v \sim 1$ and w near the positive real line is still needed since there may be removable singularity for some v. The integrals are estimated similar to the method above, and we need only a more precise approximation to T_3. Since an asymptotic expansion for T_3 is derived in the next section, we drop the details for deriving (3.5) here to avoid repetition.

5. An asymptotic expansion for $\mathbb{V}(X_{n,k})$ in the middle range. We first prove in this section the following expansion for $\mathbb{E}(X_{n,k}^2)$.

LEMMA 4. *If $\alpha \in [\![2-\sqrt{2}, 2+\sqrt{2}]\!]$, then*

$$\mathbb{E}(X_{n,k}^2) \sim n^{2(\alpha - \alpha\log(\alpha/2)-1)} \sum_{j\ge 1} \frac{\eta_j(\alpha)}{(\log n)^j}, \tag{5.1}$$

for some coefficients $\eta_j(\alpha)$.

Proof. Since

$$\mathbb{E}(X_{n,k}^2) = \mathbb{E}(X_{n,k}(X_{n,k}-1)) + O(\mathbb{E}(X_{n,k})),$$

and by the estimate (3.1) and the analysis in the last section, we need to evaluate the integral

$$T_5 := \frac{1}{2\pi i}\int_{\substack{|w|=(\alpha/2)^2\\ |\arg(w)|\le\varepsilon}} w^{-k-1} n^{4\sqrt{w}-2} \int_0^{\varepsilon} G(w,u) u^{-1/2} n^{-4\sqrt{w}u}\, \mathrm{d}u\, \mathrm{d}w,$$

where

$$G(w,u) := \frac{2w}{\pi\sqrt{2-u}(4\sqrt{w}(1-u)-2w-1)\Gamma(4\sqrt{w}(1-u)-1)}.$$

By applying Laplace's method (or Watson's lemma; see Wong (1989)) for the inner integral, we obtain

$$T_5 \sim \sum_{j\ge 0} \frac{\Gamma(j+1/2)}{(4\log n)^{j+1/2}} \cdot \frac{1}{2\pi i}\int_{\substack{|w|=(\alpha/2)^2\\ |\arg(w)|\le\varepsilon}} g_j(w) w^{-k-j/2-5/4} n^{4\sqrt{w}-2}\, \mathrm{d}w,$$

where ($\kappa(w) := 4\sqrt{w} - 2w - 1$)

$$\begin{aligned} g_j(w) &:= [u^j]G(w,u) \\ &= \frac{\sqrt{2}\,w}{\pi\Gamma(4\sqrt{w}-1)} \sum_{0\le m\le j} \frac{(4\sqrt{w})^{j-m}}{\kappa(w)^{j-m+1}} \sum_{0\le \ell\le m} \binom{2\ell}{\ell} 8^{-\ell} [u^{m-\ell}] \frac{\Gamma(4\sqrt{w}-1)}{\Gamma(4\sqrt{w}-1-4\sqrt{w}u)}. \end{aligned}$$

Then a straightforward application of the saddlepoint method leads to (5.1). Note that

$$\eta_j(\alpha) = O\left(|4\alpha - \alpha^2 - 2|^{-2j-1}\right), \tag{5.2}$$

when $\alpha \to 2 \pm \sqrt{2}$ (from inside the interval $(2-\sqrt{2}, 2+\sqrt{2})$), implying that the asymptotic expansion (5.1) is meaningful in the region $[\![2-\sqrt{2}, 2+\sqrt{2}]\!]$. □

Note that the asymptotic expansion (5.1) can also be derived in a more straightforward way by starting from (2.5) and applying the expansion for the modified Bessel function (see Abramowitz and Stegun (1965, section 9.7)).

Proof of Theorem 4. From the asymptotic expansion (3.2), we obtain

$$(\mathbb{E}(X_{n,k}))^2 \sim n^{2(\alpha-\alpha\log(\alpha/2)-1)} \sum_{j\geq 1} \frac{\xi_j(\alpha)}{(\log n)^j} \tag{5.3}$$

for some coefficients $\xi_j(\alpha)$. Then combining (5.3) and (5.1) leads to (3.13) with $\upsilon_j(\alpha) = \eta_j(\alpha) - \xi_j(\alpha)$. □

Calculations of the coefficients. The coefficients in the expansions (5.1) and (5.3) can be easily computed with the assistance of any symbolic softwares, but are very challenging by hand. For example, when $\alpha = 2 + t/\log n$, we can rewrite (3.13) as

$$\mathbb{V}(X_{n,k}) \sim n^{2(\alpha-\alpha\log(\alpha/2)-1)} \sum_{j\geq 1} \frac{p_j(t)}{(\log n)^{j+2}}, \tag{5.4}$$

where $p_j(t)$ is a polynomial of degree $j+1$ given by $p_j(t) := \sum_{0\leq \ell \leq j+1} \upsilon_{j+2-\ell}^{(\ell)}(2) t^\ell/\ell!$. Since the coefficients of $(\log n)^{-1}$ and $(\log n)^{-2}$ are both zero in the expansion, we need explicit coefficients of $\upsilon_j(\alpha)$ for $j = 1, 2, 3$ in order to get the form for $p_1(t)$.

In particular, writing $q(x) := -x^2 + 4x - 2$ and $\bar{\alpha} := 2\alpha - 1$, we have

$$\begin{aligned}
\eta_1(\alpha) &= \frac{\alpha}{2\pi q(\alpha)\Gamma(\bar{\alpha})},\\
\eta_2(\alpha) &= \frac{-1}{12\pi q(\alpha)^3\Gamma(\bar{\alpha})}\big(-6\alpha^2 q(\alpha)^2\psi'(\bar{\alpha}) + 6\alpha^2 q(\alpha)^2\psi(\bar{\alpha})^2\\
&\qquad -24\alpha(\alpha-1)q(\alpha)\psi(\bar{\alpha}) + \alpha^4 + 16\alpha^3 - 52\alpha^2 + 32\alpha + 4\big),\\
\eta_3(\alpha) &= \frac{\left(\begin{array}{l}
36\alpha^4 q(\alpha)^4\psi(\bar{\alpha})^4 + 48\alpha^3 q(\alpha)^3 q_1(\alpha)\psi(\bar{\alpha})^3\\
-12\alpha^2 q(\alpha)^2\left[18\alpha^2 q(\alpha)^2\psi'(\bar{\alpha}) - q_2(\alpha)\right]\psi(\bar{\alpha})^2\\
+48\alpha q(\alpha)\left[3\alpha^3 q(\alpha)^3\psi''(\bar{\alpha}) - 3\alpha^2 q_1(\alpha) q(\alpha)^2\psi'(\bar{\alpha}) - q_3(\alpha)\right]\psi(\bar{\alpha})\\
-36\alpha^4 q(\alpha)^4\psi'''(\bar{\alpha}) + 48\alpha^3 q(\alpha)^3 q_1(\alpha)\psi''(\bar{\alpha})\\
-12\alpha^2 q(\alpha)^2 q_2(\alpha)\psi'(\bar{\alpha}) + 108\alpha^4 q(\alpha)^4\psi'(\bar{\alpha})^2 + q_4(\alpha)
\end{array}\right)}{144\pi\alpha q(\alpha)^5\Gamma(\bar{\alpha})},
\end{aligned}$$

where ψ is the logarithmic derivative of the Gamma function and

$$\begin{aligned}
q_1(\alpha) &= \alpha^2 - 10\alpha + 8,\\
q_2(\alpha) &= \alpha^4 + 16\alpha^3 + 32\alpha^2 - 112\alpha + 76,\\
q_3(\alpha) &= 7\alpha^5 + 3\alpha^4 - 68\alpha^3 + 108\alpha^2 - 52\alpha - 4,\\
q_4(\alpha) &= \alpha^8 + 320\alpha^7 - 856\alpha^6 - 1600\alpha^5 + 8920\alpha^4 + 11264\alpha^3 + 4640\alpha^2 + 256\alpha + 16.
\end{aligned}$$

The first three $\xi_j(\alpha)$'s are given by (see (3.2))

$$\begin{aligned}
\xi_1(\alpha) &= \frac{1}{2\pi\alpha\Gamma(\alpha)^2}, \quad \xi_2(\alpha) = -\frac{6\alpha^2\psi'(\alpha) - 6\alpha^2\psi(\alpha)^2 - 1}{12\pi\alpha^2\Gamma(\alpha)^2},\\
\xi_3(\alpha) &= \frac{\left(\begin{array}{l}
-18\alpha^4\psi'''(\alpha) + 24\alpha^3(3\alpha\psi(\alpha) - 2)\psi''(\alpha)\\
+72\alpha^4\psi'(\alpha)^2 - 12\alpha(12\alpha^2\psi(\alpha)^2 - 12\alpha\psi(\alpha) + 1)\psi'(\alpha)\\
+36\alpha^4\psi(\alpha)^4 - 48\alpha^3\psi(\alpha)^3 + 12\alpha^2\psi(\alpha)^2 + 1
\end{array}\right)}{144\pi\alpha^3\Gamma(\alpha)^2}.
\end{aligned}$$

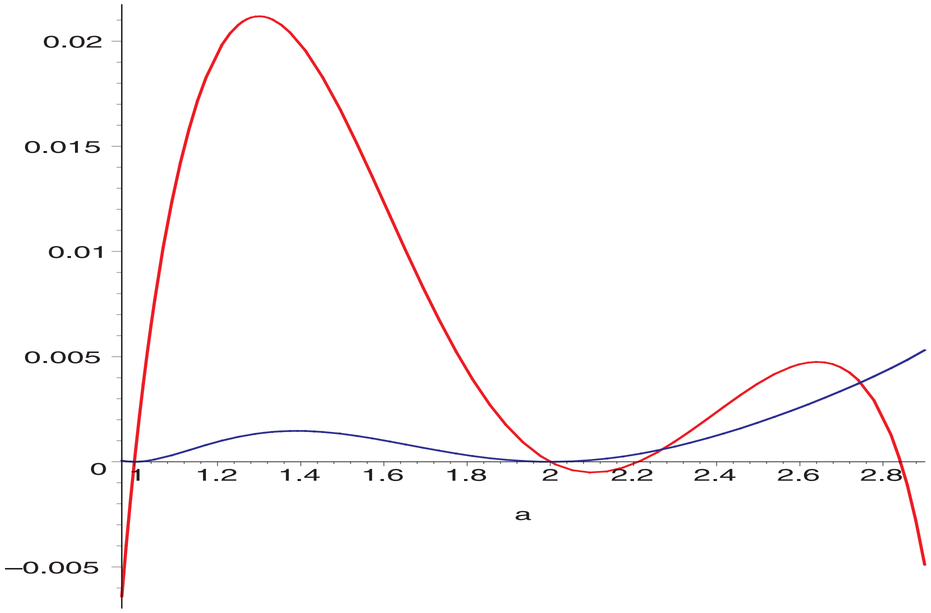


FIG. 5.1. *A plot of the two functions $\upsilon_1(\alpha)$ (smaller amplitude) and $\upsilon_2(\alpha)$. There are additional zeros for the latter besides 1 and 2, but they are minor.*

The exact forms of ξ_j and η_j are less important; the special property we need is that (see Figure 5.1)

$$\upsilon_1(i) = \upsilon_1'(i) = \upsilon_2(i) = 0 \qquad (i = 1, 2).$$

Order of the two "humps." By the expansions (5.4) and

$$2(\alpha - \alpha \log(\alpha/2) - 1) = 2 + \sum_{j \geq 2} \frac{(-1)^{j-1} t^j}{j(j-1)2^{j-2}(\log n)^j},$$

when $\alpha = 2 + t/\log n$, we have, when $t = o((\log n)^{2/3})$,

$$\mathbb{V}(X_{n,k}) \sim \frac{n^2}{720\pi(\log n)^3} e^{-t^2/(2\log n)} \left(p_1(t) + \frac{p_2(t)}{\log n} + \cdots \right).$$

Since $p_1(t)$ is a quadratic polynomial, the asymptotic maximum of the right-hand side is easily seen to be reached at $t = \pm\sqrt{2\log n} + O(1)$, and

$$\mathbb{V}(X_{n,k}) = e^{-1} \frac{n^2}{(\log n)^3} \left(\frac{21 - 2\pi^2}{24\pi} \log n \mp \frac{\sqrt{2}(21 - 2\pi^2)}{72\pi} \sqrt{\log n} + O(1) \right),$$

for $t = \pm\sqrt{2\log n} + O(1)$. This roughly explains why the left "hump" is higher than the right "hump."

Expansions for $\alpha = 1 + o(1)$ are similar.

The valley. When $k = \lfloor 2\log n \rfloor$, we have

$$\mathbb{V}(X_{n,k}) \sim \frac{p_1(\{2\log n\})}{720\pi} \cdot \frac{n^2}{(\log n)^3}.$$

Since $p_1(t)$ is concave upward, the minimum of $p_1(\{2\log n\})$ is asymptotically achieved at the subsequence of n for which $\{2\log n\} \to 1 - t_0$.

Note that the range in (3.14) where $\mathbb{V}(X_{n,k}) \geq (C + o(1))n^2/(\log n)^3$ can be extended from $O(\sqrt{\log n})$ to t_n, where $t_n \to \infty$ is given by $t_n^2 e^{-t_n^2/(2\log n)} = C$, which (expressible in terms of Lambert's W-function) asymptotically satisfies

$$t_n = \sqrt{2\log n \log\log n}\left(1 + \frac{\log\log\log n + O(1)}{\log\log n}\right).$$

6. Transitional behaviors. We prove Theorem 3 in this section. By analogy, we prove only the first two estimates (3.9) and (3.10).

The first phase transition at $3 - 2\sqrt{2}$. Recall that $D_\nu(z)$ denotes the parabolic cylinder functions (see (3.8) and Chapter 19 in Abramowitz and Stegun (1965)). Define $\beta = 3/2 - \sqrt{2}$. To describe the transitional behavior (3.9) of $\mathbb{E}(X_{n,k}^2)$ near the point $\alpha = 3 - 2\sqrt{2}$, it suffices to evaluate the integral T_4 defined in (4.8) and prove the following estimate.

LEMMA 5. *If* $\alpha = 2\beta + \sqrt{2\beta}t/\sqrt{\log n}$, *then*

$$T_4 = \frac{h(\beta)\sqrt{\beta}}{\sqrt{2\pi}}\, e^{t^2/4} D_{-1/2}(-t) k^{-1/4}\,(\alpha/2)^{-k} n^{\alpha-1}\left(1 + O\left(\frac{|t| + |t|^3}{\sqrt{k}}\right)\right) \tag{6.1}$$

uniformly for $t = o((\log n)^{1/6})$.

Estimates uniformly valid in a wider interval of α can be derived by standard tools for handling coalescence of algebraic singularities and saddlepoints; see Bleistein and Handelsman (1975). We content ourselves here with the above estimates using the following simpler method of proof.

Proof. Assume first that $\alpha < 2\beta$. By the change of variables $w \mapsto \alpha(1 + iv/\sqrt{k})/2$, we deduce that

$$\begin{aligned} T_4 = {} & h(\alpha/2)(\alpha/2)^{-k+1/2} n^{\alpha-1} k^{-1/4} \cdot \frac{1}{2\pi}\fint_{-\varepsilon\sqrt{k}}^{\varepsilon\sqrt{k}} \frac{e^{-v^2/2}}{\sqrt{\Delta\sqrt{k} - iv}}\left(1 + O\left(\frac{|v| + |v|^3}{\sqrt{k}}\right)\right) \mathrm{d}v \\ & + O\left((\alpha/2)^{-k} n^{\alpha-1-\varepsilon}\right), \end{aligned}$$

where $\Delta := 2\beta/\alpha - 1$ and $\fint$ means that an indentation (upward) of the integration path is needed if $\Delta = 0$. For the integral on the right-hand side, we use the integral representation (see Abramowitz and Stegun (1965, p. 688))

$$\frac{1}{2\pi}\fint_{-\infty}^{\infty} \frac{e^{-v^2/2}}{\sqrt{x - iv}}\,\mathrm{d}v = \frac{e^{x^2/4}}{\sqrt{2\pi}}\, D_{-1/2}(x) \qquad (x \in \mathbb{R})$$

and the estimates (3.11). The estimate (6.1) then follows by the expansion

$$\Delta\sqrt{k} = -t + O\left(t^2(\log n)^{-1/2}\right). \qquad \square$$

The second phase transition at $\alpha = 2 - \sqrt{2}$. From the proof of (5.1), we have

$$\begin{aligned} T_5 &= \frac{1}{2\pi i}\int_{\substack{|w|=(\alpha/2)^2 \\ |\arg(w)|\leq\varepsilon}} w^{-k-1} n^{4\sqrt{w}-2}\,\frac{g_0(w)\sqrt{\pi}}{\sqrt{4\sqrt{w}\log n}}\left(1 + O\left(\frac{1}{|\kappa(w)|\log n}\right)\right)\mathrm{d}w \\ &= \frac{n^{-2}}{\sqrt{\log n}} \\ &\quad \cdot \frac{1}{2\pi i}\int_{\substack{|u|=\alpha/2 \\ |\arg(u)|\leq\varepsilon}} \frac{g(u)}{u - (1 - 2^{-1/2})}\, u^{-2k} n^{4u}\left(1 + O\left(\frac{1}{|4u - 2u^2 - 1|\log n}\right)\right)\mathrm{d}u, \end{aligned}$$

where

$$g(u) := \frac{\sqrt{2u}(u-(1-2^{-1/2}))}{\sqrt{\pi}(4u-2u^2-1)\Gamma(4u-1)}.$$

We need to prove the following estimate, which implies (3.10).

LEMMA 6. *If $\alpha = 2-\sqrt{2}+\sqrt{1-2^{-1/2}}t/\sqrt{\log n}$, then*

$$T_5 = g(\alpha/2)e^{t^2/2}\Phi(-t)(\log n)^{-1/2}n^{2-\alpha-2\alpha\log(\alpha/2)}\left(1+O\left(\frac{1+|t|^3}{\sqrt{\log n}}\right)\right)$$

uniformly for $t = o((\log n)^{1/6})$.

Proof. The proof follows, *mutatis mutandis*, the same pattern as for (6.1), starting with the change of variables $v = \alpha(1+iv/\sqrt{2k})/2$. The main difference is that

$$\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{e^{-v^2/2}}{iv-x}\,\mathrm{d}v = e^{x^2/2}\Phi(-x) \qquad (x\in\mathbb{R}),$$

where the integration path has to be indented suitably downward when $x = 0$. Note that

$$g(1-2^{-1/2}) = \frac{\sqrt{2-\sqrt{2}}}{2\sqrt{2\pi}\Gamma(3-2\sqrt{2})}. \qquad \square$$

7. Profile of recursive trees. We briefly discuss the profile of random recursive trees in this section.

One way of constructing a random recursive tree of n nodes is as follows. One starts from a root node holding the key 1; at stage i ($i = 2,\dots,n$) a new node holding i is attached uniformly at random to one of the previous nodes. The process stops after node n is inserted. By construction, the values of the nodes along any path from the root to a node forms an increasing sequence. For a survey on probabilistic properties of recursive trees, see Smythe and Mahmoud (1995).

Let $Y_{n,k}$ denote the number of *internal nodes* at level k in a random recursive tree of n nodes. Then (see van der Hofstad, Hooghiemstra, and van Mieghem (2002)

$$Y_{n,k} \overset{d}{=} Y_{I_n',k-1} + X^*_{n-I_n',k},$$

where $(I_n'),(Y_{n,k}),(Y^*_{n,k})$ are independent, $Y^*_{n,k}$ is a copy of $Y_{n,k}$, and I_n' takes any of the values in $\{1,\dots,n-1\}$ with equal probability $1/(n-1)$.

From this recursive decomposition, we deduce that

$$\begin{cases} P_0(z,y) & = 1+\dfrac{yz}{1-z}, \\ P_{k+1}(z,y) & = 1+z\exp\left(\displaystyle\int_0^z \frac{P_k(t,y)-1}{t}\,\mathrm{d}t\right) \quad (k\ge 0), \end{cases}$$

where $P_k(z,y) := \sum_n \mathbb{E}(y^{Y_{n,k}})z^n$. Adopting the same set of symbols used for BSTs, we obtain

$$\sum_{n,k}\mathbb{E}(Y_{n,k})w^k z^n = z(1-z)^{-1-w},$$

so that

$$\mathbb{E}(Y_{n,k}) = \frac{s(n,k+1)}{(n-1)!}.$$

Similarly, for the second factorial moment,

$$\sum_{n,k} \mathbb{E}(Y_{n,k}(Y_{n,k}-1))w^k z^n = 2\sqrt{w}z(1-z)^{-1-w}\int_0^z (1-t)^{w-1} I_1\left(2\sqrt{w}\,\log\frac{1}{1-t}\right)\mathrm{d}t,$$

where

$$I_1(z) := \frac{1}{2}\sum_{m\geq 0}\frac{z^{2m+1}}{m!(m+1)!4^m}$$

denotes the modified Bessel function of first order. Note that, for $|w| > 4$,

$$2\sqrt{w}\int_0^1 (1-t)^{w-1} I_1\left(2\sqrt{w}\,\log\frac{1}{1-t}\right)\mathrm{d}t = \frac{4}{w\sqrt{1-4/w}(1+\sqrt{1-4/w})}.$$

The same set of tools used for BSTs also applies here; the analytic context is indeed much simpler since it is known that (see Meir and Moon (1978), van der Hofstad, Hooghiemstra, and van Mieghem (2002))

$$\mathbb{E}(Y_{n,k}^2) = \sum_{0\leq j\leq k}\binom{2j}{j}\frac{s(n,k+j+1)}{(n-1)!};$$

compare (2.6).

The asymptotic behaviors of $\mathbb{E}(Y_{n,k}^2)$ can be summarized as follows. Again let $\alpha := k/\log n$.

– If $\alpha \in [0,2]\!]$, then

$$\mathbb{E}(Y_{n,k}^2) \sim \frac{(\log n)^{2k}}{(1-\alpha/2)\Gamma(2\alpha+1)k!^2}; \tag{7.1}$$

– if $\alpha = 2 + t/\sqrt{\log n}$, then

$$\mathbb{E}(Y_{n,k}^2) \sim \frac{1}{24\sqrt{\pi}}\,\Phi(t)k^{-1/2}4^{-k}n^4,$$

uniformly for $t = o((\log n)^{1/6})$;

– if $\alpha \in [\![2,4]\!]$, then

$$\mathbb{E}(Y_{n,k}^2) \sim \frac{1}{24\sqrt{\pi(4\log n - k)}}\,4^{-k}n^4;$$

– if $\alpha = 4 + 2t/\sqrt{\log n}$, then

$$\mathbb{E}(Y_{n,k}^2) \sim \frac{1}{24\sqrt{2\pi}}\,e^{t^2/2}D_{-1/2}(t)k^{-1/4}4^{-k}n^4,$$

uniformly for $t = o((\log n)^{1/6})$;

– if $\alpha \in [\![4, K]$, then

$$\mathbb{E}(Y_{n,k}^2) \sim \left(1 + \frac{4}{\alpha\sqrt{1-4/\alpha}(1+\sqrt{1-4/\alpha})}\right) \frac{(\log n)^k}{\Gamma(\alpha+1)k!}.$$

From (7.1) and the following estimate for the mean

$$\mathbb{E}(Y_{n,k}) \sim \frac{(\log n)^k}{\Gamma(\alpha+1)k!} \qquad (\alpha \in [0, K]),$$

we obtain, for $\alpha \in [0, 2]\!]$,

$$\mathbb{V}(Y_{n,k}) \sim \varphi(\alpha) \frac{(\log n)^{2k}}{k!k!},$$

where

$$\varphi(\alpha) = \frac{1}{(1-\alpha/2)\Gamma(2\alpha+1)} - \frac{1}{\Gamma(\alpha+1)^2}.$$

The function $\varphi(\alpha)$ satisfies $\varphi(1) = \varphi'(1) = 0$, and the same type of bimodal behavior occurs when $\alpha = 1 + O(1/\sqrt{\log n})$, with the variance varying from $n^2/(\log n)^3$ to $n^2/(\log n)^2$ there. Finer results as those for BSTs can be derived; we omit all details here.

Interestingly, the bimodality of $\mathbb{V}(Y_{n,k})$ occurs when $n \geq 17$ (much smaller than that for BSTs) with the exception of $n = 21, \ldots, 32$ and $n = 64, 65, 66$.

Acknowledgment. Part of the work of the second author was finished while he was visiting Institut für Stochastik und Mathematische Informatik, J. W. Goethe-Universität (Frankfurt); he thanks the institute for hospitality and support.

REFERENCES

M. Abramowitz and I. A. Stegun (1965), *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*, Dover, New York.

D. Aldous (1991), *The continuum random tree.* II. *An overview*, in Stochastic Analysis (Durham, 1990), London Math. Soc. Lecture Note Ser. 167, Cambridge University Press, Cambridge, pp. 23–70.

D. Aldous (1996), *Probability distributions on cladograms*, in Random Discrete Structures, D. Aldous and R. Pemantle, eds., Springer, New York, pp. 1–18.

D. Aldous and P. Shields (1988), *A diffusion limit for a class of randomly-growing binary trees*, Probab. Theory Related Fields, 79, pp. 509–542.

M. T. Barlow, R. Pemantle, and E. A. Perkins (1997), *Diffusion-limited aggregation on a tree*, Probab. Theory Related Fields, 107, pp. 1–60.

F. Bergeron, P. Flajolet, and B. Salvy (1992), *Varieties of increasing trees*, in CAAP '92 (Rennes, 1992), Lecture Notes in Comput. Sci. 581, Springer, Berlin, pp. 24–48.

N. Bleistein and R. A. Handelsman (1975), *Asymptotic Expansions of Integrals*, Holt, Rinehart and Winston, New York.

G. G. Brown and B. O. Shubert (1984), *On random binary trees*, Math. Oper. Res., 9, pp. 43–65.

B. Chauvin, M. Drmota, and J. Jabbour-Hattab (2001), *The profile of binary search trees*, Ann. Appl. Probab., 11, pp. 1042–1062.

B. Chauvin, T. Klein, J.-F. Marckert, and A. Rouault (2005), *Martingales, Embedding and Tilting of Binary Trees*, to appear.

B. Chauvin and A. Rouault (2004), *Connecting Yule process, bisection and binary search tree via martingales*, J. Iranian Statistical Society, 3, pp. 89–116.

H.-H. Chern and H.-K. Hwang (2001), *Transitional behaviors of the average cost of quicksort with median-of-$(2t+1)$*, Algorithmica, 29, pp. 44–69.

L. Comtet (1974), *Advanced Combinatorics*, Revised and enlarged edition, D. Reidel Publishing Co., Dordrecht, The Netherlands.

L. Devroye (1987), *Branching processes in the analysis of the heights of trees*, Acta Inform., 24, pp. 277–298.

L. Devroye (1988), *Applications of the theory of records in the study of random trees*, Acta Inform., 26, pp. 123–130.

L. Devroye (1999), *Universal limit laws for depths in random trees*, SIAM J. Comput., 28, pp. 409–432.

L. Devroye (2003), *Limit laws for sums of functions of subtrees of random binary search trees*, SIAM J. Comput., 32, pp. 152–171.

L. Devroye and J. M. Robson (1995), *On the generation of random binary search trees*, SIAM J. Comput., 24, pp. 1141–1156.

M. Drmota and B. Gittenberger (1997), *On the profile of random trees*, Random Structures Algorithms, 10, pp. 421–451.

S. R. Finch (2003), *Mathematical Constants*, Cambridge University Press, Cambridge, UK.

P. Flajolet and A. M. Odlyzko (1990), *Singularity analysis of generating functions*, SIAM J. Discrete Math., 3, pp. 216–240.

M. Fuchs, H.-K. Hwang, and R. Neininger (2004), *Profiles of Random Trees: Limit Theorems for Random Recursive Trees and Binary Search Trees*, preprint.

G. H. Gonnet and R. Baeza-Yates, *Handbook of Algorithms and Data Structures*, 2nd ed., Addison-Wesley, Wokingham, UK, 1989.

J. M. Hammersley (1951), *The sum of products of the natural numbers*, Proc. London Math. Soc., 1, pp. 435–452.

H.-K. Hwang (1995), *Asymptotic expansions for the Stirling numbers of the first kind*, J. Combin. Theory, Ser. A, 71, pp. 343–351.

H.-K. Hwang and R. Neininger (2002), *Phase change of limit laws in the quicksort recurrence under varying toll functions*, SIAM J. Comput., 31, pp. 1687–1722.

J. Jabbour-Hattab (2001), *Martingales and large deviations for binary search trees*, Random Structures Algorithms, 19, pp. 112–127.

G. Kersting (1998), *On the Height Profile of a Conditioned Galton-Watson Tree*, preprint.

D. E. Knuth (1998), *The Art of Computer Programming, Volume* III*, Sorting and Searching*, 2nd ed., Addison-Wesley, Reading, MA.

G. Louchard (1987), *Exact and asymptotic distributions in digital and binary search trees*, RAIRO Inform. Théor. Appl., 21, pp. 479–495.

G. Louchard and W. Szpankowski (1995), *Average profile and limiting distribution for a phrase size in the Lempel-Ziv parsing algorithm*, IEEE Trans. Inform. Theory, 41, pp. 478–488.

W. C. Lynch (1965), *More combinatorial properties of certain trees*, Comput. J., 7, pp. 299–302.

H. M. Mahmoud (1992), *Evolution of Random Search Trees*, John Wiley & Sons, New York.

H. Mahmoud and B. Pittel (1984), *On the most probable shape of a binary search tree grown from a random permutation*, SIAM J. Algebraic Discrete Methods, 5, pp. 69–81.

S. N. Majumdar and P. L. Krapivsky (2003), *Extreme value statistics and traveling fronts: Various applications*, Phys. A, 318, pp. 161–170.

A. Meir and J. W. Moon (1978), *On the altitude of nodes in random trees*, Canad. J. Math., 30, pp. 997–1015.

J. Pitman (1999), *The SDE solved by local times of a Brownian excursion or bridge derived from the height profile of a random tree or forest*, Ann. Probab., 27, pp. 261–283.

B. G. Pittel (1984), *On growing random binary trees*, J. Math. Anal. Appl., 103, pp. 461–480.

R. T. Smythe and H. M. Mahmoud (1995), *A survey of recursive trees*, Theory Probab. Math. Stat., 51, pp. 1–27.

N. M. Temme (1993), *Asymptotic estimates of Stirling numbers*, Stud. Appl. Math., 89, pp. 233–243.

R. van der Hofstad, G. Hooghiemstra, and P. van Mieghem (2002), *On the covariance of the level sizes in random recursive trees*, Random Structures Algorithms, 20, pp. 519–539.

R. Wong (2001), *Asymptotic Approximations of Integrals*, SIAM, Philadelphia.

SIAM J. Discrete Math.
Vol. 19, No. 1, pp. 46–68

WHAT COSTS ARE MINIMIZED BY HUFFMAN TREES?[*]

GUNNAR FORST[†] AND ANDERS THORUP[†]

Abstract. We characterize those functions on weighted trees that are minimized at Huffman trees and those that are minimized at trees with the same level sequence as a Huffman tree. An important tool is a set of inequalities between weights of subtrees shown to be characteristic for Huffman trees. A byproduct is an algorithm transforming an arbitrary weighted tree into a Huffman tree; for a given tree, the maximal number of steps that may be taken by this algorithm is a numerical measure of how far the tree is from being a Huffman tree.

AMS subject classifications. 05C05, 05C35, 94A29

DOI. 10.1137/S0895480103421725

0. Introduction. Huffman's famous algorithm [6] constructs, for a given vector $W = (w_1, w_2, \ldots, w_n)$ of weights, a binary W-weighted tree T of minimal *weighted path length*, that is, with smallest possible value of the sum,

$$\mathrm{wpl}(T) = \sum_{i=1}^{n} w_i \ell_i;$$

here ℓ_i denotes the *level* in T of the leaf with weight w_i. A *Huffman tree* is one that can be obtained by Huffman's algorithm; the weighted path length is in fact minimized by any tree of *Huffman levels*, that is, a weighted tree having the same level sequence as a Huffman tree, and as proved by Kou [10, Theorem 3, p. 142] by no other trees.

0.1. The purpose of this paper is to analyze natural cost functions, defined on weighted trees with a given vector of weights, that are minimized at Huffman trees or at trees having some of the properties of Huffman trees. Our analysis is based on the observation that from any given weighted tree we may obtain a new tree with the relevant properties, by using certain transformations that change the structure of the given tree. Each transformation is the result of applying successively a finite number of *flips* to the tree, where a flip is an interchange of two disjoint subtrees.

Our basic idea is to single out for a given weighted tree T, not itself a Huffman tree, certain flips that are *allowed* in the sense that they change T into a tree that is "more like" a Huffman tree. Correspondingly, for us a *cost function* G is a function on weighted trees, decreasing under allowed flips. In other words, a cost function is not only minimized at Huffman trees, but its value $G(T)$ on an arbitrary weighted tree T should decrease when an allowed flip is applied to T. Intuitively, the value $G(T)$ of the cost function should indicate how much T deviates from being a Huffman tree.

It must of course be specified which flips are "allowed"; in fact, we consider several natural classes of "allowed" flips. At a minimum we require for an allowed flip that it moves the subtree with the bigger weight closer to the root. Such a flip will be called *monotonizing* (or an *m-flip* for short). More precisely, the interchange of two disjoint

[*] Received by the editors January 24, 2003; accepted for publication (in revised form) September 8, 2004; published electronically June 3, 2005. The second author was supported in part by the Danish Natural Science Research Council, grant 21–02–0458.
http://www.siam.org/journals/sidma/19-1/42172.html

[†] Matematisk Afdeling, Universitetsparken 5, DK–2100 Copenhagen, Denmark (forst@math.ku.dk, thorup@math.ku.dk).

subtrees U and V is an *m-flip* if

$$\ell_U \geqslant \ell_V \text{ and } \mathrm{w}(U) \geqslant \mathrm{w}(V);$$

here ℓ_U denotes the level and $\mathrm{w}(U)$ the total weight of the subtree U. Notice that an m-flip for which one of the two inequalities is an equality is *reversible*: in the tree resulting from the flip, the interchange of U and V is again monotonizing (resulting in the original tree). In the case of equal levels, the flip is called *horizontal* and in the case of equal weights it is called *weight neutral.* A *strict flip* is an m-flip where both inequalities are strict, and a cost function is *strict* if it is strictly decreasing under strict flips.

0.2. Clearly, the more flips that are considered as "allowed," the stronger the requirements to the corresponding cost functions. If arbitrary horizontal flips are allowed, then the corresponding cost functions depend only on the level sequence of the tree. The biggest class is the class of all m-flips. A *level cost*, by definition, is a tree function decreasing under all m-flips. For example, the weighted path length $\mathrm{wpl}(T)$ is a strict level cost.

It follows from the main theorem of section 3 that any level cost is minimized at trees of Huffman levels, and that a strict level cost is minimized at no other trees. This is an easy generalization of a result of Parker; see below. In fact, in the abstract setup described below it is natural to restrict to m-flips that interchange a subtree with its uncle; they are called *special flips.* A tree function required only to be decreasing under special flips and horizontal flips is called a *level precost.* It is a nontrivial generalization of Parker's result that it holds for the wider class of level precosts.

Allowed flips of a second type are considered in section 5. They are "adapted" to Huffman trees and they do not include arbitrary horizontal flips. In section 2 we characterize Huffman trees among weighted trees by a certain set of inequalities between weights of subtrees. *Huffman flips* (*H-flips*) are flips in a non-Huffman tree that tend (in a heuristic sense) to decrease the number of violated inequalities. Corresponding to the class of H-flips we have the notion of *Huffman costs.*

The basic examples of Huffman costs are defined from the multiset of "internal weights" $(t_1, t_2, \ldots, t_{n-1})$ of a W-weighted tree T. Hu and Tucker [5] proved that any W-Huffman tree H has "minimal" sequence of internal weights, say $(h_1, h_2, \ldots, h_{n-1})$ (ordered increasingly), in the sense that for any W-weighted tree T,

$$\sum_{k=1}^{m} h_k \leqslant \sum_{k=1}^{m} t_k, \quad \text{for} \quad m = 1, 2, \ldots, n-1.$$

This result of Hu and Tucker was a key to a theorem of Glassey and Karp [3]: Huffman trees minimize (among W-weighted trees) all tree functions of the form

$$G_f(T) = \sum_{k=1}^{n-1} f(t_k),$$

where $f(t)$ is a given nondecreasing, concave function, and conversely, a W-weighted tree T for which the value $G_f(T)$ is minimized for *all* such f is, in fact, a Huffman tree. The expression $G_f(T)$, when f is the identity function, is simply $\mathrm{wpl}(T)$. We prove the analogous minimizing property for any Huffman cost; in fact, under natural "strictness" conditions on a Huffman cost G we prove conversely that a weighted tree T minimizing G is necessarily a Huffman tree. Hence it is a consequence of our results, that if $G_f(T)$ is minimum for just one strictly increasing, strictly concave function f, then T is a Huffman tree.

0.3. The flip distance to a Huffman tree. The results in sections 3 and 5 may be used to derive numerical measures of how much a given weighted tree T deviates from being a Huffman tree: there is a maximal number $N(T)$ of strict H-flips in any sequence of H-flips that can be applied successively to T, and when that many strict H-flips are applied, then the resulting tree is a Huffman tree. So $N(T)$ may be thought of as a distance from T to a Huffman tree; the function $N(T)$ is a strict Huffman cost. In principle this result leads to an algorithm: If T is not a Huffman tree, apply a strict H-flip to T. After a finite number of steps, T is transformed into a Huffman tree. It is a consequence of the results in section 5 that this algorithm contains no loops.

Similarly, there is a maximal number of strict flips in any sequence of m-flips that can be applied to T. When that many are applied, the result is a tree of Huffman levels.

0.4. Abstract weight algebras. Huffman's algorithm depends on the way in which total weights are assigned to subtrees: the weight of a father is the sum of the weights of the two sons. Parker [12], exploiting Huffman's idea for other optimization problems, studied general types of weight combination functions, that is, binary operations on weights. This work was continued by Knuth [8]; the weights considered by Parker are nonnegative numbers, but Knuth works in an abstract framework. We will, like Knuth, work with weights taken from an abstract *weight algebra* $(\mathcal{A}, \circ, \leqslant)$, that is, an abstract set $\mathcal{A}$ with a linear order $\leqslant$ and a commutative operation $\circ$ which is *monotone* in the following sense:

(C1) $x \leqslant y \Rightarrow x \circ a \leqslant y \circ a.$

In addition to the standard assumption (C1) we will consider the following extra conditions that may be imposed on the weight algebra:

(C2) $(x \circ y) \circ (a \circ b) = (x \circ a) \circ (y \circ b)$ (bisymmetry of $\circ$).

(C3) $x \leqslant y \Rightarrow (x \circ a) \circ y \leqslant (y \circ a) \circ x$ (the associative inequality).

(C1$^+$) $x < y \Rightarrow x \circ a < y \circ a$ (strict monotonicity).

(C3$^+$) $x < y \Rightarrow (x \circ a) \circ y < (y \circ a) \circ x$ (the strict associative inequality).

Clearly, (C2) and (C3) hold if the composition is associative, that is, when the weight algebra is a commutative, linearly ordered semigroup $(\mathcal{A}, +, \leqslant)$. An important general example is based on such a semigroup: If ε is an endomorphism of $(\mathcal{A}, +, \leqslant)$ we may define a new composition $x \circ_\varepsilon y := (1 + \varepsilon)(x + y)$ for which (C1), (C2), and (C3) hold (and even (C1$^+$) and (C3$^+$) if ε is strictly increasing and (C1$^+$) holds for $+$). In fact, the basic composition $x \circ y := \lambda x + \lambda y$ considered by Parker (for $\lambda > 1$) is obtained from $(\mathbb{R}_+, +, \leqslant)$ by taking as ε multiplication by $\lambda - 1$. The compositions (1), (3), (4), and (5) considered by Knuth [8, p. 219–20] may be obtained similarly; for example (1) is obtained with $\varepsilon(x, x') = (0, x)$.

A natural cost function in this setting is the "total weight": $\mathrm{w}(T)$ is the *value* of the $\circ$-expression in the weights of T (the expression's parenthesis structure being determined by the tree structure of T). Parker proved [12, Theorem 1, p. 475] that this function is minimized by Huffman trees if natural compatibility conditions are assumed. These conditions are precisely that the total weight function is a level cost as defined above.

Parker claimed in [12, Theorem 2], that the conditions (C2) and (C3) imply that the total weight function is a level cost. Parker left most of the argument to the reader, and Knuth pointed out in [8, p. 218] that further assumptions are needed for Parker's claim. Knuth suggested in the references to [8] to replace (C1) and (C3) by their strict forms ($\mathrm{C1}^+$) and ($\mathrm{C3}^+$). The "erratum page" Parker [13] acknowledges the mistake. Apparently an argument for the corrected version of Parker's claim has not been published. We provide a proof in section 4.

Under the following extra *positivity* condition (C0) on the weight algebra: $x \leqslant x \circ y$, Knuth proved [8, pp. 217–218] that (C1), (C2), and (C3) imply that the total weight $\mathrm{w}(T)$ is minimized at Huffman trees. It is easy to see that the condition (C2) implies that the total weight $\mathrm{w}(T)$ is unchanged under horizontal flips, and obviously, (C1) and (C3) imply that $\mathrm{w}(T)$ is decreasing under special flips; hence $\mathrm{w}(T)$ is a level precost as defined above. Therefore, our general results on level precosts imply that $\mathrm{w}(T)$ is minimized at Huffman trees. In other words, the extra condition (C0) assumed by Knuth is not needed.

0.5. Contents. The main results are in sections 3 and 5. In section 3 we prove that any weighted tree, with a sequence of m-flips may be transformed into a Huffman tree; in fact, it may be done with a sequence consisting of m-flips that are either horizontal or special.

In the classical case of positive real weights it is well known that a Huffman tree is monotone: of two leaves, the one closer to the root has the bigger weight. This property may fail when weights are taken from a general weight algebra. However, it's a main step in our proof of Theorem 3.2 that any tree can be transformed into a monotone tree by a sequence of m-flips each of which is either a horizontal flip or a strict special flip. This step is taken in section 1. In fact, the definition of strict special flips requires only that a total weight is assigned to any subtree; in section 1 we allow total weights to be assigned quite arbitrarily, not necessarily as the value of the $\circ$-expression determined by the tree.

Gallager [2] gave an elegant characterization of the trees produced by Huffman's algorithm, with an application to "dynamic" Huffman coding; weights are assumed by Gallager to be nonnegative, with at most one zero weight, but Knuth [9] gave a formulation with multiple zero weights allowed. This Gallager–Knuth characterization generalizes immediately to the abstract case of nonnegative weights of a commutative ordered semigroup. The main result of section 2 is an analogous characterization of Huffman trees with weights from an abstract weight algebra. (Knuth in [8] observes that Huffman trees in absence of (C0) are "of comparatively little interest," which seems justified, although in [7] it's emphasized, in the text and several exercises, that the weights are arbitrary real.) Our characterization is in terms of a comprehensive set of inequalities among weights of subtrees, essentially, (1) that any subtree (with an uncle) has weight smaller than that of its uncle, and (2) that the two weight intervals determined by any two disjoint pairs of brother subtrees are separated (common interval ends are allowed). This characterization is fundamental to our work; section 2 also shows, with several examples, the importance of assuming the strict monotonicity ($\mathrm{C1}^+$) of the operation $\circ$.

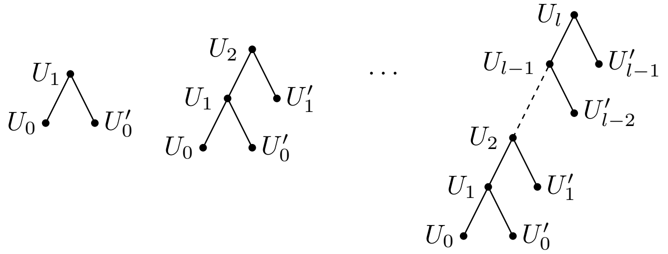
In sections 4 and 5 we assume that (C1) holds in the strict form ($\mathrm{C1}^+$); equivalently, since the order is linear, we assume that the cancellation law holds for $\circ$. In addition we assume the conditions (C2) and (C3). Section 4 describes two fundamental order relations, called *majorization order* and *Schur order*, on multisets of weights; they are intended for comparison of trees via their internal weights. In the case of a commutative ordered semigroup these two orders are natural generalizations of classical orders on multisets of reals; our generalization to arbitrary weight algebras is nontrivial.

Section 5 studies cost functions that are minimized at Huffman trees. Clearly, a horizontal flip in a Huffman tree may interfere with the separation of weight intervals for disjoint subtrees and lead to a non-Huffman tree. So the class of allowed flips must be narrowed. The relevant flips to allow in this case are the H-flips. One main result is Theorem 5.3; it implies that any weighted tree T can be transformed into a Huffman tree with a finite number of H-flips. This result seems quite analogous to Theorem 3.2 which implies that any weighted tree can be transformed into a tree of Huffman levels with a finite number of m-flips. However, our derivations of the two results are quite different. For the proof of Theorem 5.3 we use the existence of a strict Huffman cost. In fact, our second main result is the identification in Theorem 5.5 of *a natural strict Huffman cost*: the multiset of internal weights (with the Schur order on multisets).

A similar proof could have been given for Theorem 3.2 if a strict level cost exists. However, we have not been able to prove for general weight algebras satisfying (C1$^+$), (C2), and (C3) that such a cost function exists, and the proof of Theorem 3.2 is inductive instead. Such a strict level cost is necessary to obtain a "distance to a tree of Huffman levels" analogous to the "distance to a Huffman tree" in section 0.3.

Theorem 5.5, for the classical case of positive real weights, is the result of Hu and Tucker mentioned earlier. Their proof is fundamentally different: it uses the order in which the internal nodes are constructed, and for general weights the result obtained in this way is different from our result.

0.6. Trees etc. In this article, all trees will be assumed to be (finite) rooted binary trees, and, in addition, weighted. We use the familiar terminology for trees, but we work with subtrees rather than with vertices: the external vertices are called *leaves*, the internal vertices correspond to (full) *subtrees* with more than one leaf. So every proper subtree U of a tree T has a *father* and a *brother*. If the father of U is not all of T, then the brother of the father is the *uncle* of U. The *ancestor sequence* of U is the sequence $U_0, U_1, \dots$, where $U_0 = U$, and U_{i+1} is the father of U_i. The sequence of brothers of ancestors is $U'_0, U'_1, \dots$, where U'_i is the brother of U_i. In particular, $U'_0 = U'$ is the brother of U. The ancestor sequence terminates when U_l is the whole tree T, and then the sequence of brothers of ancestors ends with U'_{l-1}; the number l is the *level* of U, denoted ℓ_U; in particular, every leaf i of T has a level, denoted ℓ_i.



Throughout, we fix a multiset W of $n \geqslant 1$ weights from $\mathcal{A}$. A tree for which the multiset of leaf weights is W is called a *W-weighted tree*.

1. Monotone trees. In this section we assume that weights are taken from a given linearly ordered set $\mathcal{A}$.

Definition 1.1. A W-weighted tree T is called *monotone* if, for all leaves i and j of T, the following holds: $w_i > w_j \implies \ell_i \leqslant \ell_j$.

The multiset of n weights W may be indexed: $w_1, \dots, w_n$ in such a way that

$$w_1 \leqslant w_2 \leqslant \cdots \leqslant w_n.$$

Accordingly, the leaves of a W-weighted tree T may be labelled with the numbers $1, \dots, n$ so that w_i is the weight of the leaf i. The sequence of weights is increasing and possibly constant on certain intervals of indices. The labelling defines a corresponding sequence of levels $(\ell_1, \dots, \ell_n)$. We will mostly assume that a labelling is chosen such that the sequence of levels is decreasing on each interval where the weight sequence is constant; with this assumption, the corresponding sequence of levels is called the *level sequence* of T. Two W-weighted trees are called *level equivalent* if they have the same level sequence.

Clearly, T is monotone if and only if the level sequence of T is decreasing,

$$\ell_1 \geqslant \ell_2 \geqslant \cdots \geqslant \ell_n.$$

Definition 1.2. The notion of an m-flip, as described in the introduction, depend on the total weights assigned to subtrees of a given W-weighted tree T. Usually, and in particular in all sections except for this one, we take as the total weight the value of the parenthesized $\circ$-expression defined by the tree. However, in this section we consider an arbitrary *total weight function* $S \mapsto \mathrm{w}(S)$ from the set of (weighted) trees to the set $\mathcal{A}$ of weights. The only requirement is that the total weight of a weighted tree with only one leaf is the weight of that leaf. The composition $\circ$ plays no role in this section. Of course, the value of the total weight function on trees with only two leaves is a binary composition, but it is not assumed that the value on trees with more than two leaves is derived from this composition.

Relative to the given total weight function w, a flip interchanging two disjoint subtrees U and V is an *m-flip* if

$$\ell_U \geqslant \ell_V \text{ and } \mathrm{w}(U) \geqslant \mathrm{w}(V). \tag{1.2.1}$$

The result of the flip is a new tree. The flip is called *strict* if both inequalities in (1.2.1) are strict. If one of the inequalities is an equality then the flip is reversible: the flip of U and V in the new tree is an m-flip, resulting in the original tree.

A horizontal flip ($\ell_U = \ell_V$) is always a reversible flip. It is easily seen that two W-weighted trees T and S are level equivalent if and only if T with a sequence of horizontal flips can be transformed into S.

A *special flip* is an m-flip of a subtree U with its uncle $V := U'_1$, with the extra condition that the weight of U is at least the weight of U', that is, if

$$\mathrm{w}(U) \geqslant \mathrm{w}(U'_1) \text{ and } \mathrm{w}(U) \geqslant \mathrm{w}(U'); \tag{1.2.2}$$

it is strict, if the first inequality is strict.

We emphasize that the class of m-flips, and hence the set of trees that may be obtained from a given weighted tree by using m-flips, depends on the given total weight function. Assume for instance that the total weight function is given by $\mathrm{w}(S) = +\infty$ for any tree S with at least two leaves. Then, with a finite number of horizontal or special flips, any tree T can be transformed into a "flat" tree, that is, a tree in which leaf levels differ by at most 1. At the other extreme, if $\mathrm{w}(S) = -\infty$ for any tree S

with at least two leaves, then, with a finite number of horizontal or special flips, any tree T can be transformed into a tree consisting of a single branch, that is, a tree having only one pair of brother leaves.

The main result in this section is that with a finite number of horizontal or strict special flips any tree T can be transformed into a monotone tree. Clearly, a tree is *subtree monotone* in the sense that no strict flips exist if and only if no strict special flips exist even after horizontal interchanges. We have not been able to prove that any tree can be transformed with a finite number of horizontal or strict special flips into a subtree monotone tree.

LEMMA 1.3. *Let S be a tree, let U be a subtree at level $h \geqslant 2$, and let $U'_0, \dots, U'_{h-1}$ be the sequence of brothers of ancestors of U. Then S can be transformed with a sequence of strict special flips each of which leaves the subtrees $U, U'_0, \dots, U'_{h-1}$ intact and such that either*

(1) *the level of U in the transformed tree is decreased by 1, or*

(2) *the transformed tree is obtained by a permutation of the subtrees $U'_0, \dots, U'_{h-1}$ in the original tree such that, with a reindexing after the permutation, the following inequalities hold:*

$$\mathrm{w}(U) \leqslant \mathrm{w}(U'_1), \quad \mathrm{w}(U'_0) \leqslant \mathrm{w}(U'_1) \leqslant \cdots \leqslant \mathrm{w}(U'_{h-1}). \tag{1.3.1}$$

Proof. Assume that a tree with the property in (1) cannot be obtained with a sequence as specified.

Let $U_0, U_1, \dots, U_h$ be the ancestor sequence of U (recall that $U_0 = U$). Then U'_i is the brother of U_i and, for $i < h-1$, U'_{i+1} is the uncle of U_i and of U'_i. Let $u_i := \mathrm{w}(U_i)$ and $u'_i := \mathrm{w}(U'_i)$. The interchange of U_i and U'_{i+1} would decrease the level of U_i, and hence also the level of U. Therefore, by the assumption in the beginning of the proof, the interchange is not a strict special flip. In other words, (1.2.2), with the first inequality strict, does not hold. Hence, for $0 \leqslant i < h-1$, we have that

$$u_i \leqslant u'_{i+1} \quad \text{or} \quad u_i < u'_i. \tag{$*$}$$

If the sequence of weights $u'_0, \dots, u'_{h-1}$ is not weakly increasing, then for some $j < h-1$, we have the inequality $u'_j > u'_{j+1}$. Combined with $(*)$, for $i = j$, it follows that the interchange of U'_j and U'_{j+1} is a strict special flip. Perform the flip and replace S by the transformed tree. Note that after the flip, the ancestor sequence of U is changed and, in particular, the weights u_i may have changed. Anyway, by the initial assumption, the inequalities of $(*)$ still hold and, after the change, $u'_j < u'_{j+1}$. Therefore, after a finite number of flips, we obtain a tree where the u'_i are weakly increasing. Since $u'_0 \leqslant u'_1$, it follows from $(*)$ that $u_0 \leqslant u'_1$, that is, $\mathrm{w}(U) \leqslant \mathrm{w}(U'_1)$. Therefore, the inequalities (1.3.1) hold. □

LEMMA 1.4. *Any W-weighted tree T can be transformed into a monotone tree by a sequence of flips, each of which is either horizontal or a strict special flip.*

Proof. We consider trees into which T may be transformed with a sequence of flips as specified. We prove, by descending induction on $k = n, \dots, 2, 1$, that T may be transformed into a tree whose leaves may be labelled with the numbers $1, \dots, n$ so that the sequence of weights is increasing: $w_1 \leqslant \cdots \leqslant w_n$, and the following condition holds:

$$\ell_n \leqslant \cdots \leqslant \ell_{k+1} \leqslant \ell_j \quad \text{for} \quad j \leqslant k. \tag{$*$}_k$$

The condition $(*)$ implies that the tree is monotone. We may start the induction at $k = n$ where the condition is vacuous.

For the inductive step, we assume that $k > 1$ and that T can be transformed into a tree satisfying $(*)_k$. We replace T by the transformed tree. It suffices to prove that T can be transformed to a tree such that

(a) the levels of the leaves $k+1, k+2, \dots, n$ are unchanged and $(*)_k$ holds.

(b) $\ell_k \leqslant \ell_j$ for $j \leqslant k-1$.

We choose among the trees into which T may be transformed and such that (a) holds one for which the level of a leaf with label at most k and weight equal to w_k is the smallest possible; relabel, if necessary, so that this smallest possible level is ℓ_k. Replace T by the chosen tree.

Assume that (b) does not hold for T. Then there exists a $p < k$ such that $\ell_p < \ell_k$. The choice of the leaf with label k excludes that $w_p = w_k$. Hence, with $u := w_k$ we have the following inequalities:

$$\ell_n \leqslant \cdots \leqslant \ell_{k+1} \leqslant \ell_p < \ell_k, \quad w_p < w_k = u.$$

Set $h := \ell_k - \ell_p + 1$. Let $U = U_0$ be the subtree with the single leaf k and consider the sequences $U_0, U_1, \dots$ and $U'_0, U'_1, \dots$ of ancestors and brothers of ancestors. The common level of U_{h-1} and U'_{h-1} is ℓ_p. Therefore, with a horizontal flip we may interchange U'_{h-1} and the leaf p. So, after the flip, the leaf p is the subtree U'_{h-1}. In particular, if $u'_j := \mathrm{w}(U'_j)$, then $u'_{h-1} = w_p$.

The subtree U_h contains none of the leaves $k+1, k+2, \dots, n$, and it has the leaf k as its subtree U_0 at level h. By the choice of T, the level of U as a subtree of U_h cannot be decreased with flips as specified. Therefore, by Lemma 1.3 applied with $S := U_h$ and $U := U_0$, it is possible to transform U_h to a tree satisfying (2) of Lemma 1.3. Replace in T the subtree U_h by the transformed U_h. Then the inequalities (1.3.1) hold, that is,

$$u \leqslant u'_1, \quad u'_0 \leqslant u'_1 \leqslant \cdots \leqslant u'_{h-1}. \tag{1.4.1}$$

Now w_p is among the u'_j in the sequence, and since $w_p < u$, it follows that $w_p = u'_0$ and $u'_0 < u'_1$. It follows that the leaf p is the tree U'_0, the brother of the leaf k, and so $\ell_k = \ell_p$. In particular, the inequalities of (b) hold for $j = p$.

We claim that, in fact, all the inequalities of (b) hold. Indeed, if $\ell_k > \ell_j$ for some $j < k$, then we could repeat for the leaf j the argument above used for p. However, this time the brother U'_0 is the leaf p with weight w_p. So both w_p and w_j occur in the sequence u'_i; as $w_p < u$ and $w_j < u$ this contradicts the inequalities (1.4.1).

Therefore, $(*)_{k-1}$ holds and the proof of the induction step is complete. □

2. Huffman trees. We assume for the rest of the article that the weights are taken from a "weight algebra" $(\mathcal{A}, \circ, \leqslant)$ as described in the introduction. Notice that although we only assume condition (C1) in the main results, the strict form (C1$^+$) is necessary to get familiar properties of Huffman trees. The total weight $\mathrm{w}(T)$ of (weighted) tree T is defined as the parenthesized $\circ$-expression given by T.

DEFINITION 2.1 (Huffman trees). A *W-Huffman tree* is a W-weighted tree that may be obtained by Huffman's algorithm, as described for the abstract setup by Knuth [8, p. 217]. The notions of *unfolding* and *collapsing* explained in [1] generalize immediately to the abstract setup, and we use them repeatedly. Recall that W is a fixed multiset of n weights from the algebra $\mathcal{A}$. Assume that $n \geqslant 2$. We denote by w and w', where $w \leqslant w'$, the two smallest elements of W, and define $\overline{w} = w \circ w'$.

The multiset obtained from W by removing w and w', and adding $\overline{w}$, is denoted $\overline{W}$. Notice that $w = w'$ is not excluded, since W is a multiset; moreover, $\overline{w}$ may be equal to one or both of w and w'.

If T is a W-weighted tree then a pair of brother leaves will be called a *Huffman pair* if their weights are the minimal weights w and w' and the $\overline{W}$-weighted tree obtained from T by collapsing the pair is a $\overline{W}$-Huffman tree. So, by definition, a W-weighted tree is a W-Huffman tree if and only if it contains a Huffman pair.

In what follows, we say that two pairs of weights from the algebra, u, u' and v, v', are *separated* if there exists a weight c such that the two weights of one of the pairs are at most equal to c and the two weights of the other pair are at least equal to c.

Lemma 2.2. *Let T be a W-Huffman tree. Consider a pair of brother leaves with weights $u \leqslant u'$. Assume that there exists a Huffman pair different from the first pair. Then $u' \leqslant u \circ u'$.*

Proof. The claim is trivially true for $n = 2, 3$, and we proceed by induction on n. Consider the $\overline{W}$-Huffman tree $\overline{T}$ obtained by collapsing a Huffman pair different from the first pair. If, in $\overline{T}$, there is a Huffman pair different from the first pair, then, by induction, $u' \leqslant u \circ u'$. Otherwise, the first pair is a Huffman pair in $\overline{T}$. Then the following inequalities hold:

$$w \leqslant w' \leqslant u \leqslant u' \leqslant \overline{w} \leqslant u \circ u'. \tag{2.2.1}$$

Indeed, $w' \leqslant u$, because w, w', u, u' are four weights of W, and $u' \leqslant \overline{w}$, because $u, u', \overline{w}$ are three weights of $\overline{W}$ and u, u' are the two smallest. Finally, since $w \leqslant u$ and $w' \leqslant u'$, it follows that $\overline{w} \leqslant u \circ u'$.

In particular, the asserted inequality is a consequence of (2.2.1). □

Theorem 2.3. *Consider, for a W-weighted tree T, the following conditions on subtrees:*

(i) *The weight of a subtree having an uncle is at most the weight of the uncle.*

(ii) *The weight pairs of any two disjoint pairs of brother subtrees are separated.*

(iii) *If the weight of a son is at most the weight of the father, then the weight of the father is at most the weight of the grandfather.*

If T is a Huffman tree, then the three conditions hold. Conversely, if (i) *and* (ii) *hold, then T is a Huffman tree.*

Proof. We use the notation of the introductory section for a subtree U and its relatives. For (ii), let V, V' be brother subtrees, disjoint from U, U'.

We observe first that (iii) is a consequence of (i). Indeed, assume that $u \leqslant u_1$. If (i) holds, then $u' \leqslant u'_1$, and hence $u_1 = u \circ u' \leqslant u_1 \circ u'_1 = u_2$.

To prove the first part of the theorem, assume that T is a Huffman tree. We have to prove, for any subtree U, the following assertion:

$u' \leqslant u'_1$ and, in the setup for (ii), the weight pairs u, u' and v, v' are separated.

This assertion will be proved by induction on n. It is trivially true if $n = 2$, so we assume that $n \geqslant 3$. Consider the $\overline{W}$-Huffman tree $\overline{T}$ obtained by collapsing a Huffman pair in T. Clearly, the assertion for $\overline{T}$ yields the assertion for T when the Huffman pair can be chosen different from (U, U'), and for (ii) also different from (V, V').

So it suffices to prove the assertion in the exceptional case when U, U' is the unique Huffman pair in T. Then $\{u, u'\} = \{w, w'\}$, and, clearly, the assertion is a consequence of the following inequality, for any subtree V disjoint from the Huffman pair: $\mathrm{w}(V) \geqslant w'$. Now, this inequality is obvious, if V is a single leaf, and if V is not a single leaf, then V is an ancestor of some pair of leaves, say of weights $\tilde{v}, \tilde{v}'$.

By Lemma 2.2, $\tilde{v} \leqslant \tilde{v} \circ \tilde{v}'$. So the assumption in (iii) with the leaf of weight $\tilde{v}$ as the son is satisfied. Moreover, since (i), and hence (iii), holds for the collapsed tree $\overline{T}$ by induction, it follows that the weights of the ancestors increase. In particular, $\mathrm{w}(V) \geqslant \tilde{v} \geqslant w'$. Thus the inequality holds and the first part of the theorem has been proved.

To prove the second part, assume that (i) and (ii) hold. We say, just for the present proof, that a pair of brother leaves is *minimal* if their weights are the two minimal elements of W. Clearly, the two conditions (i) and (ii) are inherited by trees obtained by collapsing. So, by induction, it suffices to prove that there exists a minimal pair in T.

Consider a leaf of weight w. We may assume that its brother is a leaf. Indeed, if the brother of w is not a leaf, then w is a granduncle of any proper subtree of the brother. In particular, w would be a granduncle to a brother pair of leaves. That pair would, by (i), be a minimal pair, in fact, with both leaves of weight w. So assume that the brother of w is a leaf, say of weight u. If $u = w'$, then the required minimal pair is obtained. So assume that $u > w'$.

Consider now a leaf of weight w'. We may assume that its brother is a leaf. Indeed, if the brother is not a leaf, then, by arguing as in the preceding paragraph, there is in the brother subtree a pair of leaves with weights at most w'. If that pair is not a minimal pair, then the two weights are equal and equal to w'. In particular, a leaf with weight w' such that the brother is a leaf, say of weight v, has been found. By (ii), the weight pairs w, u and w', v are separated. Since $u > w'$, it follows easily that $w = w' = v$. In particular, the pair with weights w', v is minimal, as required. □

Corollary 2.4. *Let T be a W-Huffman tree. Then any tree obtained from T by collapsing a pair of brother leaves is a Huffman tree. In particular, any pair of brother leaves with weights w, w' is a Huffman pair.*

Proof. The assertion follows from Theorem 2.3. □

Corollary 2.5. *Assume* (C1$^+$). *Let T be a W-Huffman tree. Then, for two disjoint subtrees at different levels, the subtree at the smaller level has the bigger weight; in particular, T is monotone. Moreover, if $n \geqslant 2$, then w and w' are weights of a pair of brother leaves in T at the maximal level.*

Proof. The second part is a consequence of the first. Indeed, assume that T is monotone. There exists a Huffman pair in T. If there is a Huffman pair at the maximal level of T, then the claim is trivially true. Assume that there is a Huffman pair at a level which is not maximal. Consider the weights $u \leqslant u'$ of any pair of brother leaves at the maximal level; then $u' \leqslant w$ since T is monotone and $w \leqslant w' \leqslant u \leqslant u'$. Hence all four weights are equal. In particular, any pair at the maximal level is a Huffman pair.

To prove the first part, let U and V be disjoint subtrees at levels $l > k$ and weights u and v. We have to prove that $u \leqslant v$. This inequality holds by condition (i) in Theorem 2.3 if V is a brother of an ancestor of U; in particular, it holds if $k = 1$. For $k > 1$, we proceed by induction on k, and assume that the inequality holds for $k - 1$. We may assume that V is not a brother of an ancestor of U. Let $\overline{u} = u \circ u'$ and $\overline{v} = v \circ v'$ where u' and v' are the weights of the brothers of U and V. The two pairs of brother weights are separated by condition (ii) in Theorem 2.3. Therefore, if $v < u$, then also $v' \leqslant u'$; so by (C1$^+$), $\overline{v} < \overline{u}$. This is a contradiction, since $\overline{v}$ and $\overline{u}$ are the weights of the fathers of V and U at levels $k - 1$ and $l - 1$. Hence $u \leqslant v$. □

Observation 2.6. (1) Consider a tree consisting of a single branch of length ℓ, that is, with only one pair of brother leaves of weights $u \leqslant u'$ and at level ℓ. Then, by Theorem 2.3, the tree is a Huffman tree if and only if $u \leqslant u' \leqslant u'_1 \leqslant \cdots \leqslant u'_{\ell-1}$.

(2) Consider a tree consisting of two simple branches of the same length, that is, with only two pairs of brother leaves of weights u, u' and v, v' at the same level ℓ. Assume that one of the weights v, v' is strictly smaller than one of weights in the other pair. Assume (C1$^+$). Then the tree is a Huffman tree if and only if the following sets of inequalities hold:

$$\begin{Bmatrix} v \\ v' \end{Bmatrix} \leqslant \begin{Bmatrix} u \\ u' \end{Bmatrix} \leqslant \begin{Bmatrix} v_1 \\ v'_1 \end{Bmatrix} \leqslant \begin{Bmatrix} u_1 \\ u'_1 \end{Bmatrix} \leqslant \cdots \leqslant \begin{Bmatrix} u_{\ell-2} \\ u'_{\ell-2} \end{Bmatrix} \leqslant \begin{Bmatrix} v_{\ell-1} \\ v'_{\ell-1} \end{Bmatrix} \leqslant \begin{Bmatrix} u_{\ell-1} \\ u'_{\ell-1} \end{Bmatrix} ;$$

note that $v_{\ell-1} = u'_{\ell-1}$ and $v'_{\ell-1} = u_{\ell-1}$ are the weights of the two subtrees at level 1. Indeed, if the inequalities hold then by Theorem 2.3 the tree is a Huffman tree. Assume conversely that the tree is a Huffman tree. Consider the following inequalities:

$$\begin{Bmatrix} v_i \\ v'_i \end{Bmatrix} \leqslant \begin{Bmatrix} u_i \\ u'_i \end{Bmatrix} \leqslant \begin{Bmatrix} v_{i+1} \\ v'_{i+1} \end{Bmatrix} . \tag{2.6.1}$$

The last set of inequalities hold because a Huffman tree is monotone by Corollary 2.5. To prove the first it suffices, since the weight pairs are separated, to prove that one of the weights v_i, v'_i is strictly smaller than one of the weights u_i, u'_i. This strict inequality holds for $i = 0$ by hypothesis; in general, if it holds for i, then, by separation, the first inequality in (2.6.1) holds; it implies that $v_{i+1} < u_{i+1}$, and so the inequality holds for $i + 1$.

Proposition 2.7. *Assume* (C1$^+$). *Let T be a W-Huffman tree, and let U and V be disjoint subtrees of the same weight. Then the tree obtained from T by interchanging U and V is a W-Huffman tree.*

Proof. The assertion is trivial if $n = 2$. Proceed by induction, and assume $n \geqslant 3$. Collapse a Huffman pair in T to obtain $\overline{T}$. Clearly, if there is a Huffman pair inside U or V or disjoint from U and V, then the assertion for $\overline{T}$ implies the assertion for T.

So it remains to consider the case where one of the subtrees, say U, is a leaf in a Huffman pair, and where no Huffman pair is contained in V; the assertion is obvious if V is a leaf. Thus it suffices to rule out the possibility that V contains two or more leaves.

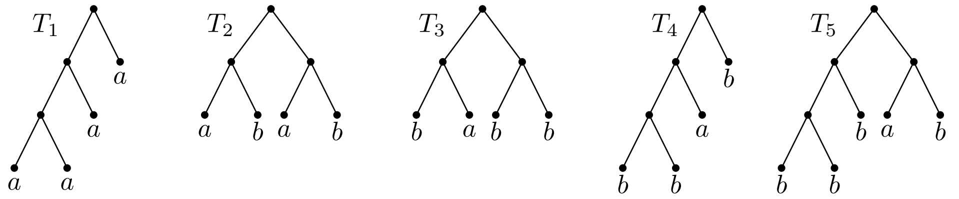
Assume, indirectly, that V contains at least two leaves. Then V contains a pair of brother leaves, say with weights $v \leqslant v'$. This pair is not a Huffman pair since no Huffman pair is contained in V. Hence, by Lemma 2.2, the weight $\overline{v} = v \circ v'$ of the father satisfies the inequality $v' \leqslant \overline{v}$. Therefore, by condition (iii) of Theorem 2.3, $\overline{v} \leqslant \mathrm{w}(V)$. Moreover, the weight of U is w or w', since U is a leaf in a Huffman pair. Hence

$$\mathrm{w}(U) \leqslant w' \leqslant v \leqslant v' \leqslant \overline{v} \leqslant \mathrm{w}(V), \tag{2.7.1}$$

and since $\mathrm{w}(U) = \mathrm{w}(V)$, it follows that all the weights in (2.7.1) are equal and equal to w'; in particular, $w' \circ w' = w'$. In addition, since the pair of brother leaves in V, both of weight w', is not a Huffman pair, it follows that $w < w'$. As a consequence, since (C1$^+$) is assumed, $w \circ w' < w' \circ w'$, that is, $\overline{w} < w'$.

It follows that the three smallest weights in $\overline{W}$ are $\overline{w}, w', w'$. Hence, in $\overline{T}$, the (collapsed) leaf with weight $\overline{w}$ is part of the Huffman pair, and so its brother is a leaf with weight w'; interchange that brother leaf with V. A contradiction is obtained: the resulting tree contains no Huffman pair, but it is a $\overline{W}$-Huffman tree by the induction hypothesis. □

Example 2.8. Consider the weight algebra with only two weights $a < b$, and $x \circ y = b$ for all x, y. Clearly (C1) holds, but (C1$^+$) does not. It's easy to see that a weighted tree with 3 leaves and weights from this weight algebra is a Huffman tree if and only if it is monotone. With 4 weights any Huffman tree is monotone; in fact, there are just 2 monotone trees with 4 weights that are not Huffman trees, those shown as T_1 and T_2 below.



The tree T_3 is clearly a Huffman tree, but T_4, which results from T_3 by the special flip of the right-hand side leaf pair with the left-most leaf, is not even of Huffman levels. The tree T_5 is a Huffman tree, but it's not monotone; the horizontal flip of the leaf pair (b, b) and the b-leaf from the leaf pair (a, b) leads to a tree which is not a Huffman tree; similarly, the flip of the left-side branch in T_5 and the b-leaf from the leaf pair (a, b) leads to a tree which is not a Huffman tree; also there is no Huffman pair in T_5 at the maximal level.

3. Minimality of trees of Huffman levels. We keep the assumptions on the weight algebra from section 2. In particular, we assume only the monotonicity condition (C1). In general, if S, U, V, ... are trees, we denote by s, u, v, ... their total weights.

Lemma 3.1. *Let T be a W-Huffman tree and let U and V be disjoint subtrees of equal weights. Then the flip of U and V can be obtained by a sequence of flips each of which is either a special flip of subtrees of equal weights or a horizontal flip.*

Proof. In the proof we say that a flip of two subtrees U and V of equal weights is a *small flip* if either $\ell_U = \ell_V$ or if $\ell_U = \ell_V + 1$ and $u \geqslant u'$. Clearly, in the second case the flip may be obtained as a horizontal flip followed by a special flip followed by the "reverse" horizontal flip.

To prove the lemma we show that the flip of U and V can be accomplished by a sequence of small flips. First, if $\ell_U = \ell_V$, then the flip is horizontal, and hence a small flip. So assume that the levels are different with, say, U at the larger level. Again, if $\ell_U = \ell_V + 1$ and $u \geqslant u'$, then the flip is a small flip.

Proceed by induction on the difference in levels, and assume either that $\ell_U = \ell_V + 1$ and $u < u'$, or that $\ell_U \geqslant \ell_V + 2$. Let S be the ancestor of U at level $\ell_V + 1$. By Theorem 2.3(i), we have the inequalities

$$u \leqslant u_1' \leqslant u_2' \leqslant \cdots \leqslant s'.$$

If $\ell_U = \ell_V + 1$, then $U = S$ and the inequalities reduce to the single inequality $u \leqslant s'$; it holds because $u < u' = s'$ in this case. Clearly, if $s' \leqslant v$, then all the inequalities above are equalities; hence we may obtain the required flip of U and V by flipping successively U and U_1', U_1' and U_2', ..., S' and V, and then reversing the first part of this sequence. Note that the inequality $s' \leqslant v$ holds if T is subtree monotone; in particular, the proof of the lemma is complete if (C1$^+$) holds; cf. Corollary 2.5.

So assume that $s' > v$. The weight pairs s, s' and v, v' are separated. So the first pair "majorizes" the second, that is, the weights of the first pair are at least equal to the weights of the second. Hence $s_1 \geqslant v_1$. If $s_1 > v_1$ or if $s_1 = v_1$ and $s_1' \geqslant s_1 = v_1 \geqslant v_1'$, then, similarly, s_1, s_1' majorizes v_1, v_1'. Moreover, if s_1, s_1' majorizes v_1, v_1', then $s_2 \geqslant v_2$. In this way we increase i as long as s_i, s_i' majorizes v_i, v_i'. When the process stops then

$$\{s_i, s_i'\} \geqslant \{v_i, v_i'\} \quad \text{and} \quad \begin{cases} \text{either } V_{i+1} \text{ is the uncle of } S_{i+1} \text{ and } s_{i+1}' \leqslant s_{i+1} = v_{i+1}, \\ \text{or } s_{i+1}' \leqslant s_{i+1} = v_{i+1} \leqslant v_{i+1}'. \end{cases}$$

Indeed, if the process stops because V_{i+1} is the uncle of S_{i+1}, then the asserted inequalities follow from Theorem 2.3(i).

Now S_{i+1} and V_{i+1} have the same weight and the difference in their levels is 1; moreover, $s_{i+1}' \leqslant s_{i+1}$. Hence the flip of S_{i+1} and V_{i+1} is a small flip. It follows from the inequalities above that the resulting tree is a Huffman tree. Moreover, in this tree the difference of levels of U and V is decreased by 2. If the original difference was 1, then it is still 1, but with the roles of U and V interchanged. Moreover, in this case $v \geqslant v'$. Therefore, in all cases, the induction hypothesis applies to the resulting tree. As a consequence, the flip of U and V in the resulting tree may be obtained by a sequence of small flips. Followed by the small flip of the trees corresponding to S_{i+1} and V_{i+1}, we obtain the flip of U and V in the original tree. So the assertion of the lemma has been proved. □

Theorem 3.2. (1) *Let T be a W-weighted tree. Then T can be transformed into a W-Huffman tree by a sequence of flips each of which is either horizontal or a strict special flip.*

(2) *Let K and H be W-Huffman trees. Then K can be transformed into H by a sequence of flips each of which is either a special flip of subtrees of equal weights or a horizontal flip.*

Proof. Clearly, the two assertions hold when $n = 1$. Proceed by induction on n. Assume that $n \geqslant 2$ and that the assertions hold for trees with $n - 1$ leaves.

To prove (1), notice that by Lemma 1.4, by a sequence of flips as specified, T may be transformed into a monotone tree. So, replacing T by the transformed tree, we may assume that T is monotone. In particular, then the two smallest weights $w \leqslant w'$ are weights of leaves at the maximal level. After a horizontal flip we may assume that w and w' are the weights of brother leaves in T. Form the collapsed tree $\overline{T}$. By the induction hypothesis, we can, with a sequence of strict and/or horizontal flips, transform $\overline{T}$ into a $\overline{W}$-Huffman tree $\overline{H}$. Clearly, these flips in the collapsed tree extend to similar flips in the original tree. Therefore, if H is the W-Huffman tree obtained by unfolding the collapsed leaf in $\overline{H}$, the sequence of extended flips transforms T into H.

To prove (2), chose a Huffman pair in K and one in H and consider the corresponding collapsed trees $\overline{K}$ and $\overline{H}$. By the induction hypothesis, we can, using a sequence of flips as specified transform $\overline{K}$ into $\overline{H}$. Under this transformation, the collapsed leaf with weight $\overline{w}$ in $\overline{K}$ corresponds to a leaf of weight $\overline{w}$ in $\overline{H}$, but not necessarily to the collapsed leaf in $\overline{H}$. So, the sequence obtained by extending these flips transforms the W-Huffman tree K into a W-Huffman tree $\widetilde{H}$ which is either equal to H or can be transformed into H by a flip of the unfolded pair with the collapsed leaf. By Lemma 3.1 the latter flip can be obtained by a sequence of flips as specified. □

Definition 3.3. *Level costs.* A function $T \mapsto G(T)$ from the set of W-weighted trees to a partially ordered set is called a *level cost* if $G(T) \geqslant G(S)$ when S results from

T by an m-flip; and G is called a *level precost* if G is only assumed to be decreasing under horizontal flips and special flips. Notice that horizontal and weight neutral flips are reversible m-flips. Hence, equivalently, G is a level cost if the following three conditions hold when S results from T by a flip of two subtrees:

(α) $G(T) = G(S)$ if the flip is horizontal,
(β) $G(T) = G(S)$ if the flip is weight neutral, and
(γ) $G(T) \geqslant G(S)$ if the flip is a strict flip.

A level cost G is said to be *strict*, if the inequality in (γ) is always strict. A level precost is *strict* if the inequality in (γ) is strict when the strict flip is an flip of a subtree U and its uncle U_1' where $\mathrm{w}(U) > \mathrm{w}(U_1')$.

COROLLARY 3.4. *Let $T \mapsto G(T)$ be a level precost. Then the value $G(S)$ on a W-weighted tree S having the same level sequence as a W-Huffman tree is minimum, that is, for any W-weighted tree T we have the inequality $G(T) \geqslant G(S)$.*

Moreover, if G is a strict level precost, then the following conditions on a W-weighted tree S are equivalent:

(i) *Any tree obtained from S by a sequence of horizontal interchanges is subtree monotone, that is, for two disjoint subtrees at different levels, the subtree at the smaller level has the bigger weight.*
(ii) *S has the same level sequence as a W-Huffman tree.*
(iii) *$G(S)$ is minimum.*

Proof. To prove the first assertion, let S and T be W-weighted trees such that S has the same level sequence as a W-Huffman tree H. As noted in Definition 1.2, there is a sequence of horizontal flips transforming H into S. By Theorem 3.2, there is a sequence of flips, each of which is either horizontal or a special flip, transforming T into H. As each such flip decreases the G-value it follows that $G(T) \geqslant G(S)$.

To prove the equivalence, we note that the implication (i)⇒(ii) holds in general, that is, without the level cost being strict. Indeed, assume that S has the property in (i). Then, in particular, S is monotone and so the weights $w \leqslant w'$ appear at the maximal level. With a flip of weights at the maximal level, we may assume that they appear as weights of a pair of brother leaves. Collapse the pair to obtain $\overline{S}$. Clearly, the condition (i) is inherited to $\overline{S}$. So, by induction, $\overline{S}$ has the same level sequence as a $\overline{W}$-Huffman tree. Therefore, S has the same level sequence a W-Huffman tree.

Next, the implication (ii)⇒(iii) follows from the first assertion of the corollary.

Finally, we prove the implication (iii)⇒(i). Assume that G is strict and that $G(S)$ is minimum. The value $G(S)$ is unchanged under horizontal flips. Hence it suffices to prove that S is subtree monotone. Consider two disjoint subtrees U and V, of weights u and v, with $\ell_U < \ell_V$. We have to prove that $u \geqslant v$. Clearly, if $\widetilde{S}$ is the tree obtained by collapsing U and V into two leaves of weights u and v, then every flip of subtrees in $\widetilde{S}$ yields a corresponding flip in S. In particular, the sequence of flips (including horizontal flips) making $\widetilde{S}$ monotone, provided by Lemma 1.4, contains no strict flips since any strict flip would decrease the value $G(S)$ strictly. Therefore $\widetilde{S}$ is itself monotone. Hence $u \geqslant v$, and the proof is complete. □

Example 3.5. *The total weight as a level cost.* Assume that the weight algebra $\mathcal{A}$ satisfies the conditions (C2) and (C3) (recall that (C1) is always assumed to hold). Then the total weight $\mathrm{w}(T)$ is a level precost. Indeed, assume that S is obtained from T by a flip as in Definition 3.3. The equality $\mathrm{w}(T) = \mathrm{w}(S)$ in (β) is obvious. The inequality $\mathrm{w}(T) \geqslant \mathrm{w}(S)$ in (γ) for a strict special flip follows from the associative inequality (C3). Finally, the equality $\mathrm{w}(T) = \mathrm{w}(S)$ in (α) follows from (C2), as proved by Knuth [8, p. 217]. So, Corollary 3.4 contains Knuth's result [8, p. 218], but here proved without Knuth's "positivity assumption" (C0).

It follows from the results of section 4, cf. 4.3(3), that if (C1$^+$) hold then $\mathrm{w}(T)$ is in fact a level cost.

If the strict conditions (C1$^+$) and (C3$^+$) hold, then $\mathrm{w}(T)$ is strict. Indeed, it follows from (C1$^+$) that if the weight of a leaf of a tree is strictly increased, in particular, if a subtree is replaced by a subtree of strictly bigger weight, then the total weight is increased. Therefore, it follows from (C3$^+$) that the inequality in (γ) is strict when the flip is strict.

Definition 3.6. *Level costs based on internal weights.* Assume that the weight algebra $\mathcal{A}$ satisfies the conditions (C2) and (C3).

Important tree functions are obtained as follows: For any weighted tree T denote by $\omega(T)$ the multiset of *internal weights* of T, that is, the set of total weights of subtrees with at least two leaves. If T has n leaves, then $\omega(T)$ is a multiset of $n-1$ weights.

If G is a symmetric function in $n-1$ weight variables, with values in a partially ordered set R, then we define a corresponding tree function,

$$G_{\mathrm{tree}}(T) := G(\omega(T));$$

in particular, if R is a commutative, partially ordered group and $g : \mathcal{A} \to R$ is any function, there is a corresponding tree function defined by

$$G_g(T) := \sum_{x \in \omega(T)} g(x),$$

where the sum is over the internal weights of T.

Clearly, G_{tree} is a level cost if G is monotone in each variable and "balanced" in the following sense:

$$G(u \circ a, v \circ b, \dots) = G(u \circ b, v \circ a, \dots),$$

and G_{tree} is strict if, in addition, (C1$^+$) holds and G is strictly monotone. Indeed, assume that S results from T as in Definition 3.3 by a flip, say of U and V. If the flip is weight neutral, then $\omega(T) = \omega(S)$. If the flip is special: V is the uncle of U and $v \leqslant u$, then the internal weights that are changed are the weights u_i for $i = 1, 2, \dots$. Of these, the first, $u_1 = u \circ b$, is decreased, since $u \geqslant v$, the second is decreased by the associative inequality, and hence those of smaller levels are decreased; as a consequence $G_{\mathrm{tree}}(T) \geqslant G_{\mathrm{tree}}(S)$. If the conditions on G are strict and the flip is strict then the decrease in u_1 is strict and so $G_{\mathrm{tree}}(T) > G_{\mathrm{tree}}(S)$. Finally, if the flip is horizontal, the equality $G_{\mathrm{tree}}(T) = G_{\mathrm{tree}}(S)$ follows from the argument of Knuth as in Definition 3.5, as G is balanced.

In particular, the tree function G_g is a level cost if

$$u \leqslant v \implies g(u) \leqslant g(v),$$

$$g(u \circ a) + g(b \circ v) = g(v \circ a) + g(b \circ u).$$

Moreover, if (C1$^+$) holds and g is strictly increasing, then G_g is a strict level cost.

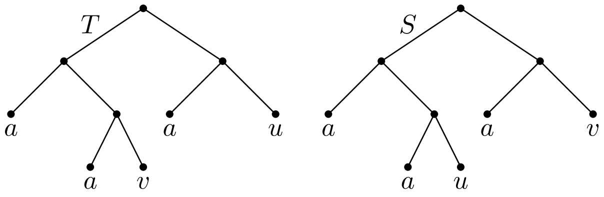
If the weight algebra itself is a stable subset of a commutative ordered group (in additive notation), we may take as g the identity. Then the corresponding tree function is the *weighted path length*,

$$G(T) = \sum_{x \in \omega(T)} x = \sum_i \ell_i w_i,$$

where the first sum is over the internal weights and the second sum is over the leaf weights. Consequently, by Corollary 3.4, a W-weighted tree S is of Huffman levels if and only if it minimizes the weighted path length.

Example 3.7. *Comments.* The equivalence of (i) and (ii) in Corollary 3.4 is proved under the assumption that a strict level precost exists. It does not hold in general, since a Huffman tree is not necessarily monotone, see Example 2.8.

If a strict level precost exists, then (C1$^+$) holds. Indeed, consider the following trees:



Here S may be obtained from T by interchanging two leaves with weights u and v, and also by interchanging two subtrees with leaves a, u and a, v; assume that $v > u$ and hence that $a \circ v \geqslant a \circ u$. Then both flips are m-flips; the first is a composition of a special strict flip and two horizontal flips. Hence, if a strict level precost G exists, we have that $G(T) > G(S)$. Therefore, the second flip is irreversible. Thus $a \circ v > a \circ u$.

The authors have tried in vain to prove that if (C1$^+$), (C2), and (C3) hold, then a strict level precost exists. Under these conditions it follows from Examples 3.5 and 3.6 that a strict level precost exists if the composition is associative or if the associative inequality holds in the strict form (C3$^+$).

4. Orders on multisets.

In this section we assume for the weight algebra $\mathcal{A}$ conditions (C1$^+$), (C2), and (C3).

Definition 4.1. Let $X = (x_1, \dots, x_m)$ be a sequence of m weights. As we do not assume that the composition is associative, we have to make a choice in the definition of the composition:

$$x_1 \circ \cdots \circ x_m := (x_1 \circ \cdots \circ x_{m-1}) \circ x_m;$$

it is the total weight of the tree $\Sigma(x_1, \dots, x_m)$ defined inductively: $\Sigma(x_1)$ is the tree with one leaf, of weight x_1, and $\Sigma(x_1, \dots, x_m)$ is the join $\Sigma(x_1, \dots, x_{m-1}) \cup \Sigma(x_m)$.

If Y is a second sequence of weights, we write $X \preccurlyeq Y$ if Y and X have the same number of elements, $Y = (y_1, \dots, y_m)$, and

$$x_1 \circ \cdots \circ x_i \leqslant y_1 \circ \cdots \circ y_i \quad \text{for } i = 1, \dots, m.$$

This relation, on the set of finite sequences of weights, is a partial order: it is obviously reflexive and transitive, and asymmetry follows from (C1$^+$). Clearly the relation depends on both the composition and the order in the weight algebra.

Lemma 4.2. *If $x_1 \circ \cdots \circ x_m \leqslant y_1 \circ \cdots \circ y_m$ and $x_1 \leqslant y_1$, then, for all weights z,*

$$z \circ x_1 \circ \cdots \circ x_m \leqslant z \circ y_1 \circ \cdots \circ y_m. \tag{4.2.1}$$

Proof. The two weights in (4.2.1) are the total weights of the two trees,

$$S := \Sigma(z, x_1, \dots, x_m) \quad \text{and} \quad T := \Sigma(z, y_1, \dots, y_m).$$

For a join of trees, we have $\mathrm{w}(S \cup U) = \mathrm{w}(S) \circ \mathrm{w}(U)$. Hence, to prove the inequality $\mathrm{w}(S) \leqslant \mathrm{w}(T)$ it suffices, by (C1$^+$), to prove the inequality $\mathrm{w}(S \cup U) \leqslant \mathrm{w}(T \cup U)$ for some fixed tree U. We take for U any tree having at level $m-1$ a leaf of weight u_1 such that $x_1 \leqslant u_1 \leqslant y_1$. As a simple choice we take the following:

$$U := \Sigma(u_1, \dots, u_m), \quad \text{with } x_1 \leqslant u_1 \leqslant y_1,$$

and with arbitrary weights u_i for $i \geqslant 2$.

Consider the join of S and U,

$$S \cup U = \Sigma(z, x_1, \dots, x_m) \cup \Sigma(u_1, \dots, u_m). \tag{4.2.2}$$

In this tree the weights x_2 and u_1 appear as weights of leaves at the same level (equal to m). So, by bisymmetry (C2), they can be interchanged without altering the total weight. After the flip, u_1 is the weight of the uncle of the leaf with weight x_1, and $x_1 \leqslant u_1$. Therefore, by the associative inequality (C3), the interchange of these two weights gives a weak increase of total weight. In the resulting tree, there is a subtree consisting of brother leaves of weights z and u_1. By bisymmetry (C2), we can interchange this subtree and the leaf of weight x_2 (both at level m) without altering the total weight. The interchange gives the following tree:

$$\Sigma(x_1, \dots, x_m) \cup \Sigma(z, u_1, \dots, u_m). \tag{4.2.3}$$

Now, in this tree, replace the subtree $\Sigma(x_1, \dots, x_m)$ by $\Sigma(y_1, \dots, y_m)$; the result is a weak increase in the total weight. Finally, use the inverses of the previous flips to transform the tree (4.2.3) into $T \cup U$. The total weight is further weakly increased, since $u_1 \leqslant y_1$. Therefore, $\mathrm{w}(S \cup U) \leqslant \mathrm{w}(T \cup U)$, and the proof is complete. □

Consequences 4.3. Let $X = (x_1, \dots, x_m)$ and $Y = (y_1, \dots, y_m)$ be sequences of weights. Then,

(1) It is an easy consequence of the lemma that insertion of a weight z at a given position in the sequences preserves the order

$$X \preccurlyeq Y \implies (x_1, \dots, x_r, z, x_{r+1}, \dots, x_m) \preccurlyeq (y_1, \dots, y_r, z, y_{r+1}, \dots, y_m).$$

(2) By repeated application of (1), it follows that "shuffling" with a given sequence $Z = (z_1, \dots, z_p)$ preserves the order

$$X \preccurlyeq Y \implies Z \vee X \preccurlyeq Z \vee Y.$$

The $\vee$-notation indicates the shuffled sequences: Z is shuffled into X and into Y with respect to a fixed strictly increasing map $\{1, \dots, p\} \to \{1, \dots, p+m\}$.

(3) Clearly, if $a \leqslant u$, then $(a, u) \preccurlyeq (u, a)$. Hence, as a special case of (2), it follows for any sequence $(z_1, \dots, z_p)$ that

$$a \leqslant u \implies (a, z_1, \dots, z_p, u) \preccurlyeq (u, z_1, \dots, z_p, a). \tag{4.3.1}$$

By combining the bisymmetry (C2) with (4.3.1) it follows that if we have two weights in a tree and the bigger weight is at the bigger level, then the flip of the two weights results in a weak decrease in the total weight. This result was announced by Parker in [12], with the weak form (C1) of (C1$^+$), and corrected in [13].

Definition 4.4. *Majorization order.* If $\mathcal{X}$ and $\mathcal{Y}$ are multisets of weights, we write $\mathcal{X} \preccurlyeq \mathcal{Y}$ if the two multisets have the same number of elements, and, when the elements are indexed increasingly, we have that

$$(x_1, \dots, x_m) \preccurlyeq (y_1, \dots, y_m).$$

For multisets of real weights, this relation $\preccurlyeq$ on multisets is the *opposite* of the weak supermajorization order denoted $\prec^w$ in [11, Formula (12), p. 10]. In spite of this conflict we will here use the name *majorization order* for the relation $\preccurlyeq$. Notice that (C1$^+$) is required for the asymmetry of the relation $\preccurlyeq$.

Let $(x_1, \dots, x_m)$ be a sequence of weights such that $x_i > x_{i+1}$ for some i. Then it follows from (C1) and (C3) that interchanging x_i and x_{i+1} gives a sequence which is smaller with respect to $\preccurlyeq$. Consequently, among all permutations of $(x_1, \dots, x_m)$, the one in which the x_i appear in weakly increasing order is smallest. From this observation and consequence (2) above, it follows that, for all multisets $\mathcal{Z}$,

$$\mathcal{X} \preccurlyeq \mathcal{Y} \implies \mathcal{Z} \cup \mathcal{X} \preccurlyeq \mathcal{Z} \cup \mathcal{Y}. \tag{4.4.1}$$

Hence, multisets with the majorization order form a partially ordered semigroup.

Definition 4.5. *Schur order.* A second order on multisets, called *Schur order* and denoted $\leqslant_s$, is defined as follows: We write $\mathcal{X} \leqslant_s \mathcal{Y}$ if $\mathcal{X}$ can be transformed into $\mathcal{Y}$ by a finite number of operations each of which consists in either replacing a weight x by a weight y, where $x \leqslant y$, or replacing weights x, u by weights y, v, where $x \leqslant u$, $x \leqslant y \leqslant v$ and $x \circ u = y \circ v$. Equivalently, the relation $\leqslant_s$ is the smallest relation on multisets which is reflexive, transitive, and compatible with union and such that

$$\begin{aligned} &\{x\} \leqslant_s \{y\} \quad \textit{if } x \leqslant y, \qquad \textit{and} \\ &\{x, u\} \leqslant_s \{y, v\} \quad \textit{if } \{x, u\} \preccurlyeq \{y, v\} \quad \textit{and} \quad x \circ u = y \circ v. \end{aligned}$$

It follows from (4.4.1) that majorization order $\preccurlyeq$ is finer than Schur order $\leqslant_s$,

$$\mathcal{X} \leqslant_s \mathcal{Y} \implies \mathcal{X} \preccurlyeq \mathcal{Y}. \tag{4.5.1}$$

In particular, from (4.5.1) we obtain the nontrivial fact that the relation $\leqslant_s$ is asymmetric (and hence an order, not merely a preorder).

Symmetric functions $G(x_1, \dots, x_m)$ in m weight variables correspond to functions $G(\mathcal{X})$ on multisets of m elements. Assume that such a function G takes values in a partially ordered set R. Then G is called *Schur concave* if, for all multisets $\mathcal{Z}$ with $m-2$ elements,

$$G(\mathcal{Z} \cup \{x, u\}) \leqslant G(\mathcal{Z} \cup \{y, v\}), \quad \textit{when } \{x, u\} \preccurlyeq \{y, v\} \textit{ and } x \circ u = y \circ v,$$

and *strictly Schur concave* if the inequality is strict whenever $\{x, u\} \neq \{y, v\}$.

Clearly G is increasing in each variable and Schur concave if and only if $G(\mathcal{X}) \leqslant G(\mathcal{Y})$ when $\mathcal{X} \leqslant_s \mathcal{Y}$; a similar assertion holds for the strict version.

A function $g(x)$ in a single weight variable x, with values in a partially ordered semigroup $(R, +, <)$, defines a symmetric function G_g in m variables: $G_g(\mathcal{X}) := \sum g(x_i)$. The function G_g is Schur concave if and only if

$$g(x) + g(u) \leqslant g(y) + g(v) \quad \textit{when } x \leqslant u,\ x \leqslant y \leqslant v,\ x \circ u = y \circ v. \tag{4.5.2}$$

Call g *concave*, if (4.5.2) holds.

In the classical case, where the weight algebra is a commutative ordered group (in additive notation), g is concave if, for all weights x, y, Δ with $x < y$ and $\Delta > 0$,

$$g(x + \Delta) - g(x) \geqslant g(y + \Delta) - g(y). \tag{4.5.3}$$

Example 4.6. *Notes.* For the case where the weight algebra is the set of positive elements (or the set of all elements) in a commutative (linearly) ordered group, it is a theorem, essentially due to Hardy, Littlewood, and Pólya, that the two partial orders $\preccurlyeq$ and $\leqslant_s$ on multisets agree; cf. [4, Lemma 2, p. 47] or [11, Lemma B.1, p. 21].

The two orders differ in general. Consider for instance real weights with addition and with Parker's "λ-composition" $x \circ_\lambda y := \lambda x + \lambda y$ (for $\lambda > 1$), mentioned in the Introduction. Denote by $\preccurlyeq_\lambda$ the majorization order with respect to the composition $\circ_\lambda$. It is easy to see that

$$\mathcal{X} \preccurlyeq \mathcal{Y} \implies \mathcal{X} \preccurlyeq_\lambda \mathcal{Y}.$$

The implication is a bi-implication for $m \leqslant 2$, and so the two compositions induce the same Schur order on multisets. But the implication is not a bi-implication for $m \geqslant 3$; in particular, the Schur order and the majorization order induced by $\circ_\lambda$ are different.

Clearly, for real weights and real functions, the condition (4.5.3) holds for an increasing function g if and only if g is concave in the usual sense. It is a result of Schur that a differentiable symmetric function G in m variables is Schur concave if and only if $(x_i - x_j)\big(\frac{\partial G}{\partial x_i} - \frac{\partial G}{\partial x_j}\big) \leqslant 0$.

A classical example of a Schur concave function in the real case is the following: Let $h_1, \dots, h_m$ be functions of a positive real variable and form the symmetric function

$$H(x_1, \dots, x_m) = \sum h_{\sigma_1}(x_1) \cdots h_{\sigma_m}(x_m),$$

where the sum is over all permutations $\sigma_1, \dots, \sigma_m$ of $1, 2, \dots, m$.

With positive numbers $t_1, \dots, t_m$ and $h_i(x) := t_i^x$ it is easily seen that the function $G := -H$ is Schur concave; it is increasing if $0 < t_i \leqslant 1$ for all i, and strictly increasing and strictly Schur concave if at least two t_i's are different.

5. Minimality of Huffman trees. We keep the assumptions on the weight algebra from section 4.

Definition 5.1. *Huffman flips.* Consider subtrees of a given tree T. An interchange of two disjoint subtrees U and V is called an *H-flip* if the following inequalities hold:

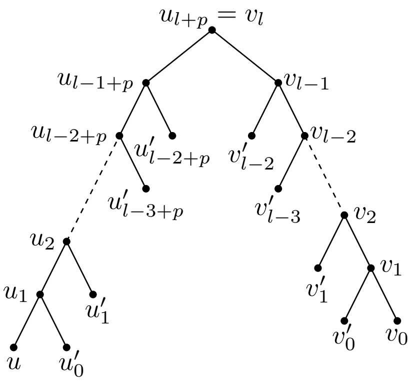
$$\ell_U \geqslant \ell_V, \quad u \geqslant v, \tag{5.1.1}$$

$$u'_i \leqslant v'_i \quad \textit{for} \quad i = 0, \dots, l{-}2. \tag{5.1.2}$$

As usual, u and v are the weights of U and V, and the u'_i and v'_i are the weights of the brothers of ancestors of U and V. The number l is the length of the path from V to the first common ancestor of U and V.

The H-flip is said to be *strict* if $u > v$ and either $\ell_U > \ell_V$ or one of the inequalities in (5.1.2) is strict. Note that an H-flip is strict if and only if it is nonreversible, that is, if and only if in the tree resulting from the H-flip, the interchange of V and U is not an H-flip.

As usual, we denote by $U_0 = U, U_1, \dots$ and $V_0 = V, V_1, \dots$ the ancestor sequences. Thus V_l is the first common ancestor of V and U, and $V_l = U_{l+p}$ for some $p \geqslant 0$. It is not excluded that $l = 1$. In this case V is a brother of an ancestor of U, in fact $V = U'_p$; and the flip is an H-flip if $u \geqslant v$, since the set of inequalities (5.1.2) is empty.



Lemma 5.2. *Let T be a tree which is not a Huffman tree. Then there are subtrees U and V in T such that the flip of U and V is a strict H-flip. In fact, the subtrees may be chosen so that either V is the uncle of U, $v < u$, and $u' \leqslant u$, or the following inequalities hold:*

$$\ell_V + 1 \geqslant \ell_U \geqslant \ell_V, \quad u > v, \quad u' < v', \quad u' \leqslant u, \quad v \leqslant v', \quad u \leqslant u'_1, \tag{5.2.1}$$

$$\left\{\begin{matrix} v \\ v' \end{matrix}\right\} \leqslant \left\{\begin{matrix} u_1 \\ u'_1 \end{matrix}\right\} \leqslant \left\{\begin{matrix} v_1 \\ v'_1 \end{matrix}\right\} \leqslant \left\{\begin{matrix} u_2 \\ u'_2 \end{matrix}\right\} \leqslant \cdots \leqslant \left\{\begin{matrix} v_{\ell-2} \\ v'_{\ell-2} \end{matrix}\right\} \leqslant \left\{\begin{matrix} u_{\ell-1} \\ u'_{\ell-1} \end{matrix}\right\} \leqslant \left\{\begin{matrix} v_{\ell-1} \\ v'_{\ell-1} \end{matrix}\right\}. \tag{5.2.2}$$

Proof. Consider the conditions (i) and (ii) in Theorem 2.3. It follows from the theorem that one of them is violated for T. If (i) is violated, then there is a subtree U such that for the uncle of U, say V, we have that $v < u$. Clearly, then, the interchange of U and V is a strict H-flip. It may be assumed in addition that $u' \leqslant u$, since otherwise the interchange of U' and V would be a strict H-flip for which the additional inequality holds.

So assume that condition (i) holds for T. Then condition (ii) is violated. As a consequence, there are two pairs of disjoint brother subtrees, say U, U' and V', V, such that the weight pairs are not separated. Choose the two pairs such that, first, ℓ_V is smallest possible, and, among those, ℓ_U is minimum. Then $\ell_U \geqslant \ell_V$. In addition, we may assume the following inequalities:

$$u' \leqslant u,\ v \leqslant v' \text{ and } v < u,\ u' < v';$$

indeed, the first two hold after a possible renaming of brothers, and then the last two are a consequence, because the weight pairs are not separated.

In particular, then (5.1.1) holds. Consider the smallest subtree containing U and V. In the notation of Definition 5.1 it is $U_{l+p} = V_l$. It follows from the minimality of the choice that if we collapse in this tree all of the subtrees $U, U', U'_1, \dots$ and $V, V', V'_1, \dots$ into single leaves, then the resulting tree satisfies the conditions of Theorem 2.3 except that the weight pairs u, u' and v', v are not separated.

Therefore, if the two brother leaves u, u' are collapsed into a single leaf, the collapsed tree is a Huffman tree. The collapsed tree has two pairs of brother leaves, of weights u_1, u'_1 and v, v'. Now $u \leqslant u_1$, since (i) holds for T, and $v < u$. Hence $v < u_1$. It follows first, since the collapsed tree is monotone by Corollary 2.5, that $\ell_V \geqslant \ell_U - 1$. Hence (5.2.1) holds. Next, since the weight pairs of the collapsed tree are separated and $v < u_1$, we obtain the first inequality in (5.2.2).

Assume first that $\ell_U = \ell_V + 1$. Apply Observation 2.6 to the collapsed tree. It follows that all the inequalities in (5.2.2) hold, because the first one holds.

Assume next that $\ell_U = \ell_V$. In the collapsed tree we may further collapse the pair of leaves with weights v, v'. Apply Observation 2.6 to this doubly collapsed tree. By symmetry, we may interchange the roles of the u's and the v's. So, we may assume that all the inequalities of (5.2.2) hold except for the first; as the first holds, they all hold. ∎

Theorem 5.3. *Any W-weighted tree T can be transformed into a W-Huffman tree by a sequence of strict H-flips. If H and K are W-Huffman trees, then H can be transformed into K by a sequence of reversible H-flips.*

Proof. For the second assertion it suffices, by (2) of Theorem 3.2, to prove for a W-Huffman tree H that the interchange of two subtrees U and V (disjoint, and not brothers) of the same weight is an H-flip, and this follows from the separation property of Theorem 2.3(ii), as in the proof of Lemma 5.2.

Consider now the first assertion. By Lemma 5.2, if T is not a Huffman tree, then it admits a strict H-flip. Therefore, to finish the proof, it suffices to show that there is a tree function which is strictly decreased by any strict H-flip. Such a tree function, called a *strict Huffman cost* in Definition 5.4, is described in Theorem 5.5. When its existence is established, the proof of the theorem is complete. ∎

Definition 5.4. *Huffman costs.* A function $T \mapsto G(T)$, from the set of weighted trees to some partially ordered set, is called a *Huffman cost* if it is decreasing under H-flips, that is,

$$G(T) \geqslant G(\widetilde{T}) \tag{5.4.1}$$

whenever $\widetilde{T}$ results from T by an H-flip. The Huffman cost G is called *strict*, if the inequality (5.4.1) corresponding to any strict H-flip is strict.

Theorem 5.5. *The function $T \mapsto \omega(T)$, associating with a W-weighted tree its multiset of internal weights* (*with the Schur order on multisets*), *is a strict Huffman cost.*

Proof. Let $\widetilde{T}$ be the result of applying an H-flip to T, interchanging two subtrees U and V as in Definition 5.1. Equivalently, $\widetilde{T}$ is obtained from T by replacing U by $\tilde{U} := V$ and V by $\tilde{V} := U$. We have to prove that $\omega(T) \geqslant \omega(\widetilde{T})$. In the notation of Definition 5.1, the flip changes the following internal weights of T:

$$u_i \text{ into } \tilde{u}_i \quad \text{for } i = 1, 2, \dots, \quad \text{and} \quad v_i \text{ into } \tilde{v}_i \quad \text{for } i = 1, \dots, l-1.$$

Hence the multiset $\omega(\widetilde{T})$ can be obtained from $\omega(T)$ by changing $\{u_i, v_i\}$ to $\{\tilde{u}_i, \tilde{v}_i\}$ for $i = 1, \dots, l-1$ and $\{u_i\}$ to $\{\tilde{u}_i\}$ for $i \geqslant l$. Therefore, to prove the assertion, it suffices to verify the following relations of multisets:

$$\{u_i, v_i\} \succcurlyeq \{\tilde{u}_i, \tilde{v}_i\} \text{ and } u_i \circ v_i = \tilde{u}_i \circ \tilde{v}_i \quad \text{for } i = 1, \dots, l{-}1, \tag{5.5.1}$$

$$\{u_i\} \succcurlyeq \{\tilde{u}_i\} \quad \text{for } i = l, l{+}1, \dots, \tag{5.5.2}$$

and to show that if the H-flip is strict, then at least one of the inequalities in (5.5.1) or (5.5.2) is strict.

Assume first that $1 \leqslant i < l$. Then $u_i \circ v_i$ is the total weight of the tree obtained as the join of U_i and V_i, containing U and V, respectively, at the same level, and $\tilde{u}_i \circ \tilde{v}_i$ is the total weight of the tree obtained from the first by interchanging U and V. Therefore, the equation $u_i \circ v_i = \tilde{u}_i \circ \tilde{v}_i$ follows from bisymmetry (C2). Moreover, since $v \leqslant u$ and $u'_j \leqslant v'_j$ for $j = 0, \dots, i-1$, we have the inequalities,

$$v \circ u'_0 \circ \cdots \circ u'_{i-1} \leqslant \left\{ \begin{matrix} u \circ u'_0 \circ \cdots \circ u'_{i-1} \\ v \circ v'_0 \circ \cdots \circ v'_{i-1} \end{matrix} \right\} \leqslant u \circ v'_0 \circ \cdots \circ v'_{i-1}.$$

In particular, $\tilde{u}_i$ is the smaller of $\tilde{u}_i, \tilde{v}_i, u_i, v_i$. Therefore, (5.5.1) holds.

Next, the subtree $\widetilde{U}_i$ for $i < l+p$ is obtained from U_i by replacing U by V. So $u_i \geqslant \tilde{u}_i$ for any $i < l+p$. If $i \geqslant l+p$, then the subtree U_i contains U and V, and u_i is the total weight of U_i. In this case, the decrease in weight, $u_i \geqslant \tilde{u}_i$, is a consequence of Lemma 4.2, as noted in 4.3(3). Thus the inequalities in (5.5.2) hold.

Assume that the H-flip is strict. Then $v < u$ and, if $p = 0$, then $u'_j < v'_j$ for some $j = 0, \dots, l-2$. Clearly, since $v < u$, we obtain the strict inequality $u_i > \tilde{u}_i$ for $i = 1, \dots, l+p-1$. Hence, if $p > 0$, then the inequality in (5.5.2), for $i = l$, is strict. If $p = 0$, then we obtain the strict inequality $\tilde{u}_i < v_i$ for $i > j$; in particular, then the inequality in (5.5.1) is strict for $i = l-1$. □

Note 5.6. Theorem 5.5 establishes the existence of a strict Huffman cost, and so completes the proof of Theorem 5.3.

Assume that G is a Huffman cost. Obviously, if $\widetilde{T}$ results from T using a reversible H-flip, then $G(T) = G(\widetilde{T})$. In particular, by the second part of Theorem 5.3, if H and K are W-Huffman trees, then $G(H) = G(K)$. Consequently, by the first part, G is minimized on the W-Huffman trees; if G is strict, then conversely, if $G(T)$ is minimum, then T is a W-Huffman tree.

Corollary 5.7. *If a symmetric function G in $n-1$ weight variables is increasing and Schur concave, then the tree function G_{tree} is a Huffman cost, and it is a strict Huffman cost if and only if G is strictly increasing and strictly concave.*

Let g be a function in a single weight variable. If g is increasing and concave, then the corresponding tree function G_g is a Huffman cost. If g is strictly increasing and strictly concave, then G_g is a strict Huffman cost.

Proof. This is an immediate consequence of Theorem 5.5. □

Note 5.8. Consider the case of real weights and a strictly increasing function $g(x)$. If g is strictly concave in the usual sense, then (4.5.3) holds. Hence, by Example 4.5, the corollary applies to tree function G_g. Consequently, the function G_g is minimized exactly at the Huffman trees. In fact, it is not difficult to see that G_g is minimized exactly at Huffman trees if and only if (4.5.3) holds in the following restricted form:

$$g(x+\Delta) - g(x) \geqslant g(y+\Delta) - g(y) \quad \text{when } 0 < x < y,\ \Delta > 0,\ y+\Delta \leqslant 2x.$$

The latter condition holds if and only if $g(x)$ is strictly concave on the positive reals; it requires no condition on the values on negative weights.

Note 5.9. As noted in Example 4.6, the Schur order on real multisets induced by Parker's λ-composition $x \circ_\lambda y = \lambda x + \lambda y$ is equal to the usual Schur order, and hence equal to the usual majorization order.

Consequently, if a symmetric function G in $n-1$ real variables is increasing and Schur concave, then the corresponding tree function $G(\omega_\lambda(T))$, where the multiset $\omega_\lambda(T)$ of internal weights is computed with respect $\circ_\lambda$, is minimized at the "λ-Huffman trees." If G is strictly increasing and strictly Schur concave, then the tree function $G(\omega_\lambda(T))$ is minimized only at the λ-Huffman trees; cf. [12, Theorem 5, p. 478].

REFERENCES

[1] G. Forst and A. Thorup, *Minimal Huffman trees*, Acta Inform., 36 (2000), pp. 721–734.

[2] R. G. Gallager, *Variations on a theme by Huffman*, IEEE Trans. Inform. Theory, 24 (1978), pp. 668–674.

[3] C. R. Glassey and R. M. Karp, *On the optimality of Huffman trees*, SIAM J. Appl. Math., 31 (1976), pp. 368–378.

[4] G. H. Hardy, J. E. Littlewood, and G. Pólya, *Inequalities*, Cambridge University Press, 1934.

[5] T. C. Hu and A. C. Tucker, *Optimal computer search trees and variable-length alphabetical codes*, SIAM J. Appl. Math., 21 (1971), pp. 514–532.

[6] D. A. Huffman, *A method for the construction of minimum-redundancy codes*, Proc. IRE, 40 (1952), pp. 1098–1101.

[7] D. E. Knuth, *The Art of Computer Programming, Volume I/Fundamental Algorithms*, 2nd ed., Addison Wesley, Reading, 1973.

[8] D. E. Knuth, *Huffman's algorithm via algebra*, J. Combin. Theory Ser. A, 32 (1982), pp. 216–224.

[9] D. E. Knuth, *Dynamic Huffman coding*, J. Algorithms, 6 (1985), pp. 163–180.

[10] L. T. Kou, *Minimum variance Huffman codes*, SIAM J. Comput., 11 (1982), pp. 138–148.

[11] A. W. Marshall and I. Olkin, *Inequalities: Theory of majorization and its applications*, Mathematics in Science and Engineering 143, Academic Press, New York, 1979.

[12] D. S. Parker, *Conditions for optimality of the Huffman algorithm*, SIAM J. Comput., 9 (1980), pp. 470–489.

[13] D. S. Parker, *Erratum: Conditions for optimality of the Huffman algorithm*, SIAM J. Comput., 27 (1998), p. 317.

SIAM J. Discrete Math.
Vol. 19, No. 1, pp. 69–82

EXACT MINIMUM DENSITY OF CODES IDENTIFYING VERTICES IN THE SQUARE GRID*

YAEL BEN-HAIM† AND SIMON LITSYN†

Abstract. An identifying code C is a subset of the vertices of the square grid $\mathbb{Z}^2$ with the property that for each element v of $\mathbb{Z}^2$, the collection of elements from C at a distance of at most one from v is nonempty and distinct from the collection of any other vertex. We prove that the minimum density of C within $\mathbb{Z}^2$ is $\frac{7}{20}$.

Key words. identifying code, square grid, graph, density

AMS subject classifications. 05C70, 68R10, 94B99, 94C12

DOI. 10.1137/S0895480104444089

1. Introduction.

Let C be a subset of the vertices of the square grid $\mathbb{Z}^2$. For a vertex $v \in \mathbb{Z}^2$ define its identifying set $I(v)$ as the set of all the elements from C coinciding or connected by an edge with v. If for all vertices of $\mathbb{Z}^2$ the identifying sets are nonempty and distinct, then C is called an identifying (ID) code. The problem is to find the minimum density of C within $\mathbb{Z}^2$. This problem was introduced in [16] in relation to fault diagnosis in arrays of processors. Here the nodes of an identifying code correspond to controlling processors able to check themselves and their neighbors. Thus the identifying property guarantees location of a faulty processor from the set of "complaining" controllers.

Bounds on the density of ID codes in $\mathbb{Z}^2$ were given in [5, 6, 16]. The best known upper bound is $\frac{7}{20}$. It was shown in [5] and is given, e.g., by a configuration depicted in Figure 1, shifted by vectors $(10a + b, 4b)$, $a, b, \in \mathbb{Z}$. The best known lower bound was $\frac{15}{43}$ (see [6] and [20]). Thus there was a gap of about 0.0012 between the upper and lower bound.

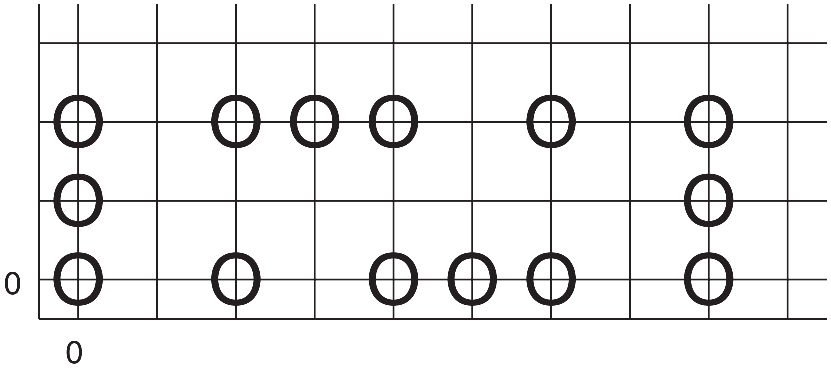


Fig. 1. *Configuration yielding an ID code with density* $\frac{7}{20}$.

*Received by the editors May 24, 2004; accepted for publication (in revised form) November 8, 2004; published electronically June 16, 2005.
http://www.siam.org/journals/sidma/19-1/44408.html

†Department of Electrical Engineering - Systems, Tel Aviv University, Tel Aviv 69978, Israel (yaelm@eng.tau.ac.il, litsyn@eng.tau.ac.il).

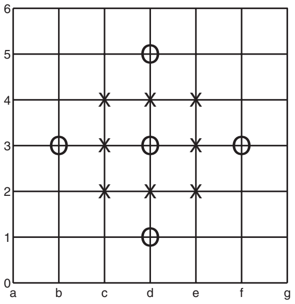


FIG. 2. *A vertex in C'.*

In the current paper we close the gap by showing that the upper bound is indeed tight, as conjectured in [5].

THEOREM 1. *The minimum density of ID codes in $\mathbb{Z}^2$ is $\frac{7}{20}$.*

Our approach is a development of the method suggested earlier in [6]. It relies on construction of a bipartite graph characterizing the relations between vertices in $\mathbb{Z}^2$ and elements of C in their environments. We tried to keep our notation similar to that of [6].

For papers dealing with ID codes in graphs other than $\mathbb{Z}^2$ and under generalizations of the ID property, we refer to [1, 2, 5, 3, 4, 6, 7, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22] and references therein.

2. Definitions. Let C be an identifying code. We address the vertices of C as *codewords*. Here V is the set of vertices in $\mathbb{Z}^2$. In what follows, we treat C and V as if they were finite; the problem with the infinity of $|C|$ and $|V|$ can be easily resolved by defining the corresponding notions in a finite torus in $\mathbb{Z}^2$ of growing size and considering limits (see [6, section 2.1]).

For $i = 1, \dots, 5$, we denote

$$L_i = \{v \in V : |I(v)| = i\}, \quad l_i = |L_i|,$$

$$L_{\geq i} = \bigcup_{j=i}^{5} L_j, \quad l_{\geq i} = |L_{\geq i}|.$$

Furthermore, we partition the set L_3 into two subsets:

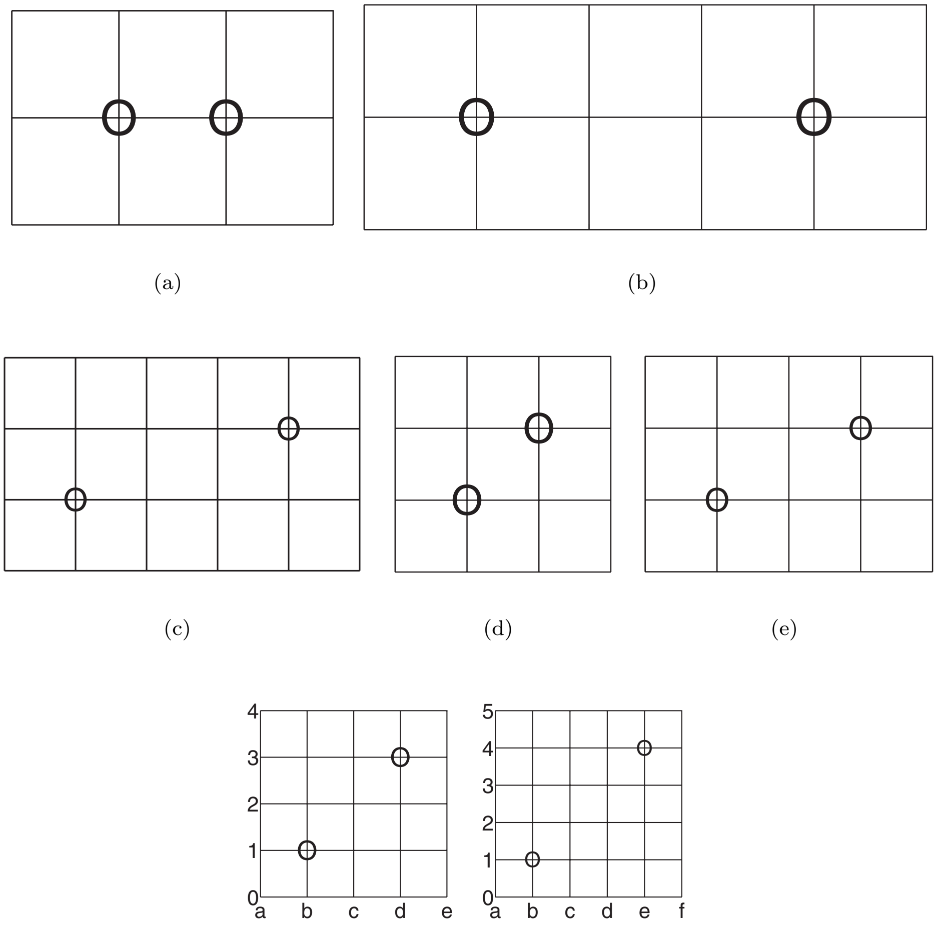
$$\widetilde{L_3} = \{v \in L_3 : \text{there exists a vertex } v' \in L_{\geq 3} \text{ such that } |I(v) \cap I(v')| = 2\}$$

and $\overline{L_3} = L_3 \setminus \widetilde{L_3}$, $\widetilde{l_3} = |\widetilde{L_3}|$, $\overline{l_3} = |\overline{L_3}|$, $\widetilde{l_3} + \overline{l_3} = l_3$.

Let us partition C into subcodes C' and C''. We define C' to be the following set of codewords:

$$C' = \{c \in C : \text{for each } v \text{ such that } c \in I(v), |I(v)| \leq 2\}$$

and $C'' = C \setminus C'$. The surroundings of a codeword in C' are shown in Figure 2. The notation in the figures throughout the paper is as follows: o is a codeword, x is a noncodeword, and unmarked vertices could be either. If $(\alpha, \beta) \in C'$, then $(\alpha + 2, \beta)$, $(\alpha - 2, \beta)$, $(\alpha, \beta + 2)$, $(\alpha, \beta - 2) \in C$, and all eight neighbors, including the diagonal ones, of (α, β) do not belong to C (in Figure 2, $\alpha = d$ and $\beta = 3$).



(a) (b)

(c) (d) (e)

(f) (g)

FIG. 3. *Impossible relations between codewords in C'.*

Figure 3 shows impossible relations between codewords in C'. In every subfigure there are two marked vertices; if both of them belong to C', then either this fact contradicts Figure 2 or the code is not an identifying code. For example,

- in Figure 3(f), if $b1, d3 \in C'$, then $|I(c2)| = 0$;
- in Figure 3(g), if $b1, e4 \in C'$, then $I(c2) = I(d3)$.

The following equality is obtained by counting in two ways the number of pairs $(c, v),\ c \in I(v)$:

$$\begin{aligned} 5|C| &= \sum_{i=1}^{5} i l_i \\ &= l_1 + 2(|V| - l_1 - l_{\geq 3}) + 3l_{\geq 3} + l_4 + 2l_5. \end{aligned} \tag{1}$$

Substituting $l_1 \leq |C|$ into (1), we get

$$6|C| \geq 2|V| + l_{\geq 3} + l_4 + 2l_5. \tag{2}$$

For C'', we have the following lemma.

LEMMA 1.

$$2\widetilde{l_3} + 3\overline{l_3} + 4l_4 + 5l_5 \geq |C''|. \tag{3}$$

Proof. We partition $\widetilde{L_3}$ into two subsets,

$$\begin{aligned}\widetilde{L_3}^{1} &= \{v \in \widetilde{L_3} : \exists v' \in \widetilde{L_3} \text{ s.t. } |I(v) \cap I(v')| = 2\},\\ \widetilde{L_3}^{2} &= \{v \in \widetilde{L_3} : \forall v' \in L_{\geq 3},\ |I(v) \cap I(v')| = 2 \Rightarrow v' \notin \widetilde{L_3}\}\end{aligned}$$

We note that $C'' = A_1 \cup A_2 \cup A_3$, where

$$\begin{aligned}A_1 &= \{c \in C'' : \exists v \in L_{\geq 3} \setminus \widetilde{L_3} \text{ s.t. } c \in I(v)\},\\ A_2 &= \{c \in C'' : \exists v \in \widetilde{L_3}^{1} \text{ s.t. } c \in I(v)\},\\ A_3 &= \{c \in C'' : \exists v \in \widetilde{L_3}^{2} \text{ s.t. } c \in I(v), \text{ and } \forall v' \in L_{\geq 3} \setminus \widetilde{L_3},\ c \notin I(v')\}.\end{aligned}$$

For each pair $v_1, v_2 \in \widetilde{L_3}^{1}$ such that $|I(v_1) \cap I(v_2)| = 2$, there are four codewords in C'' which belong to the identifying sets of v_1 or v_2 (i.e., $|I(v_1) \cup I(v_2)| = 4$). Hence, $|A_2| \leq 2|\widetilde{L_3}^{1}|$. For each vertex $v \in \widetilde{L_3}^{2}$ there is at most one codeword in C'' which belongs to $I(v)$ but not to the identifying set of any vertex in $L_{\geq 3} \setminus \widetilde{L_3}$; hence $|A_3| \leq |\widetilde{L_3}^{2}|$. Therefore, $|A_2 \cup A_3| \leq 2\widetilde{l_3}$, and since $|A_1| \leq 3\overline{l_3} + 4l_4 + 5l_5$, the claim follows. □

3. What we do. For an identifying code C, we construct in section 4 a bipartite graph Γ whose vertex set is $C' \cup L_{\geq 3}$ (i.e., each edge is in $C' \times L_{\geq 3}$) such that the degree of every element of C' is at least 4, the degree of every element of $\overline{L_3}$ is at most 2, the degree of every element of $\widetilde{L_3}$ is at most 6, and the degree of every element of $L_{\geq 4}$ is at most 4. A vertex from a subset satisfying the corresponding bound on its degree is said to have a *legal degree.*

Before we proceed with the construction, we present an argument which leads from the existence of Γ to the bound $|C| \geq \frac{7}{20}|V|$.

The existence of Γ implies that

$$2\overline{l_3} + 6\widetilde{l_3} + 4l_{\geq 4} \geq \text{the number of edges in } \Gamma \geq 4|C'|; \tag{4}$$

hence,

$$\overline{l_3} + 3\widetilde{l_3} + 2l_{\geq 4} \geq 2|C'|. \tag{5}$$

We denote $l_4 = \alpha l_{\geq 3}$ and $l_5 = \beta l_{\geq 3}$; then by (3) and (2),

$$3l_{\geq 3} + l_4 + 2l_5 \geq |C''| + \widetilde{l_3},$$

$$l_{\geq 3} \geq \frac{1}{3 + \alpha + 2\beta}\left(|C''| + \widetilde{l_3}\right). \tag{6}$$

Plugging (6) into (2), we get

$$6|C| \geq 2|V| + \frac{1 + \alpha + 2\beta}{3 + \alpha + 2\beta}\left(|C''| + \widetilde{l_3}\right),$$

$$6|C| \geq 2|V| + \frac{1}{3}\left(|C''| + \widetilde{l_3}\right). \tag{7}$$

Hence, by (2) + 6(7) we get

$$42|C| \geq 14|V| + \overline{l_3} + 3\widetilde{l_3} + 2l_4 + 3l_5 + 2|C''|. \tag{8}$$

We substitute (5) into (8) to get

$$42|C| \geq 14|V| + 2(|C'| + |C''|) + l_5. \tag{9}$$

Therefore,

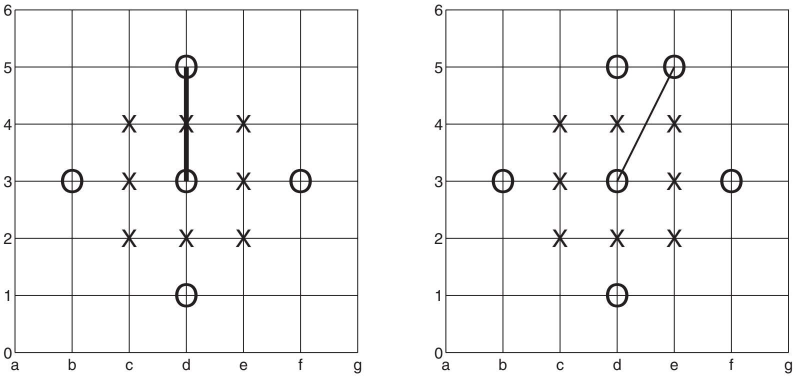
$$40|C| \geq 14|V|. \tag{10}$$

4. The construction of Γ. The construction consists of 10 steps. The graph obtained after the ith step is denoted Γ_i; hence $\Gamma = \Gamma_{10}$. The vertex set of Γ_i for each i is $C' \cup L_{\geq 3}$. Every step adds new edges to Γ. After each step, before we proceed to the next step, we show that all the elements of $L_{\geq 3}$ have legal degrees in Γ_i (Lemmas 2–11). In other words, we show that the increments in the degrees of the elements of $L_{\geq 3}$ do not go beyond their legal degrees. After the last step, we show that the degrees of all the elements of C' are at least 4 (Lemmas 12–17) and conclude that all the degrees in Γ are legal (Corollary 4.1).

In most of the steps there are figures, which describe configurations of codewords, along with rules about how to add edges to Γ. The same instructions indeed should be applied also in the rotations and reflections of the figures.

Throughout the rest of the paper, we often use Figures 2 and 3 to determine that a certain element of $\mathbb{Z}^2$ cannot belong to C'. However, for the sake of fluency we omit the references to these figures.

Step 1. Construct the bipartite graph Γ_1 with no edges whose vertex set is $C' \cup L_{\geq 3}$.

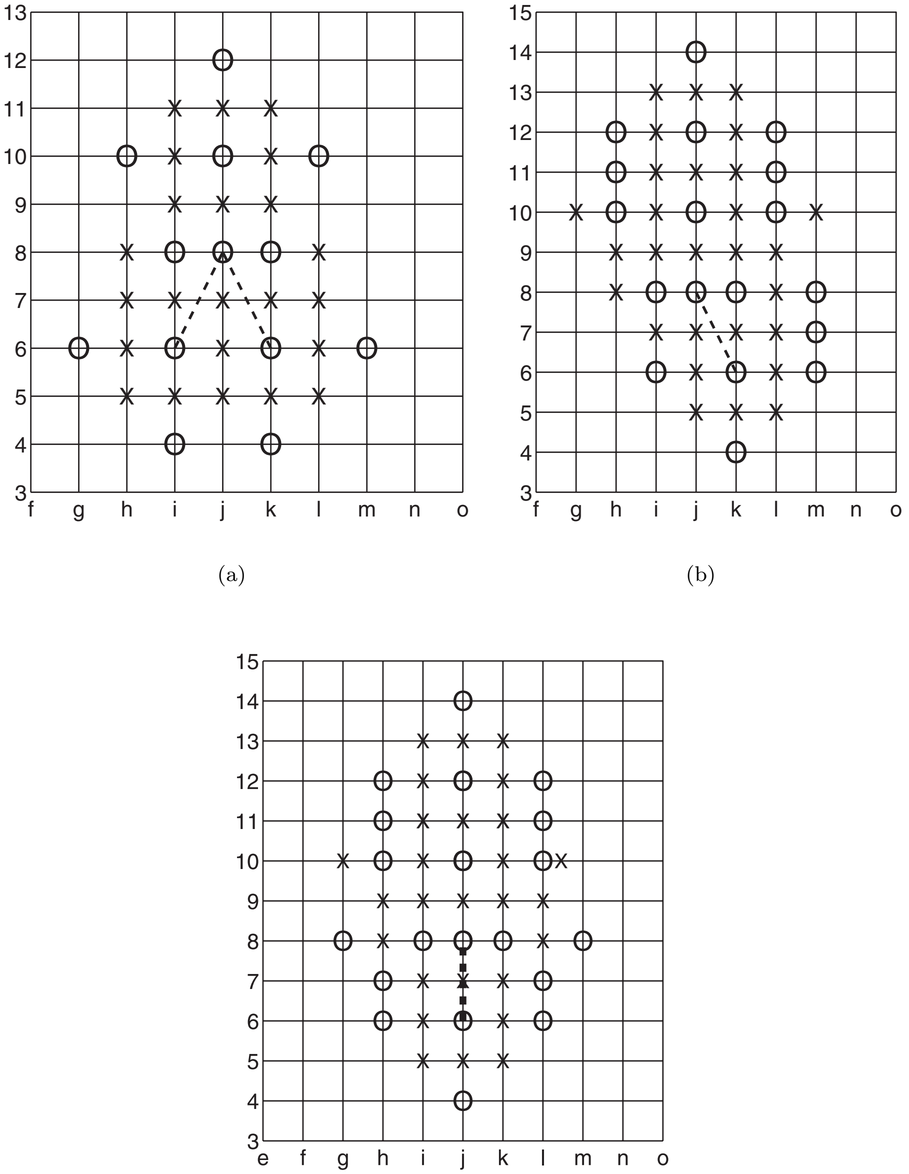


(a)

(b)

FIG. 4. *Step* 2.

Step 2. Add the edge $(d3, d5)$ marked in Figure 4(a) if $d5 \in L_{\geq 3}$, and the edge $(d3, e5)$ marked in Figure 4(b) if $e5 \in L_{\geq 3}$, unless it is either the case in Figure 5(a), where the edges $(k6, j8)$ and $(i6, j8)$ should not be added, or the case in Figure 5(b), where the edge $(k6, j8)$ should not be added, or the case in Figure 5(c), where the edge $(j6, j8)$ should not be added.



(a)

(b)

(c)

Fig. 5. *The marked edges are not added in Step* 2.

The following lemmas are straightforward (they are proved in [6]; notice that Figure 5(a) is used only in Lemma 2, and Figures 5(b) and 5(c) are used only at a later stage in this paper).

Lemma 2. *The degree of each element of* $L_{\geq 3} \setminus \widetilde{L_3}$ *in* Γ_2 *is at most* 2.

Lemma 3. *The degree of each element of* $\widetilde{L_3}$ *in* Γ_2 *is at most* 3.

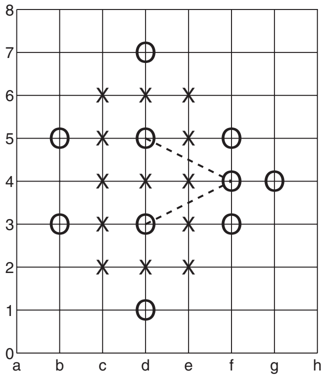


Fig. 6. *The marked edges are not added in Step* 3.

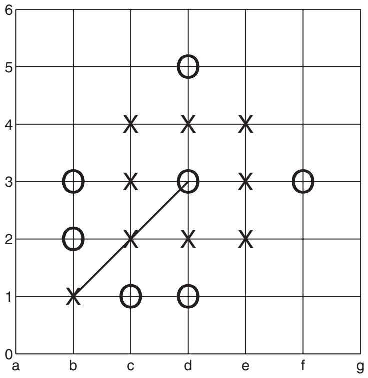


Fig. 7. *Step* 4.

Step 3. For each edge (c_1, c_2) in Γ_2, $c_1 \in C'$, $c_2 \in \widetilde{L_3} \cup L_{\geq 4}$, add an edge (c_1, c_2) (i.e., there are two edges between c_1 and c_2) unless it is the case in Figure 6, where the edges $(d3, f4), (d5, f4)$ should not be added.

Lemma 4. *All the elements of $L_{\geq 3}$ have legal degrees in Γ_3.*

Proof. By Lemma 3, the degree of an element of $\widetilde{L_3}$ in Γ_2 is at most 3; therefore its degree in Γ_3 is at most 6. By Lemma 2, the degree of an element of $L_{\geq 4}$ in Γ_2 is at most 2; therefore its degree in Γ_3 is at most 4. □

Step 4. Add the edge marked in Figure 7, i.e., the edge $(d3, b1)$ ($b1 \in L_{\geq 3}$ since $I(b1) \neq I(c2)$).

Lemma 5. *All the elements of $L_{\geq 3}$ have legal degrees in Γ_4.*

Proof. In Γ_3 the degree of $b1$ is 0 (since $b1 \notin C$). If $b_1 \in L_4$, i.e., $a1, b0 \in C$, then $b1$ can have at most four neighbors in Γ_4. If $b1 \in L_3$, i.e., either $a1 \in C$ or $b0 \in C$, then $b1$ can have at most two neighbors in Γ_4. □

Step 5. Add the edge marked in Figure 8, i.e., the edge $(h3, e3)$.

Lemma 6. *All the elements of $L_{\geq 3}$ have legal degrees in Γ_5.*

Proof. If $d3 \notin C$, then $h3$ is the only neighbor of $e3$ in Γ_5. If $d3 \in C$, then $e3 \in L_4$ can have neighbors in $b3$ (Step 5) and in $c1, c5$ (Step 4), the number of which is at most four (in fact at most three: if $c1$ and $c5$ are connected to $e3$, then $c3 \in C$ and $b3$ cannot be connected to $e3$ by Step 5). □

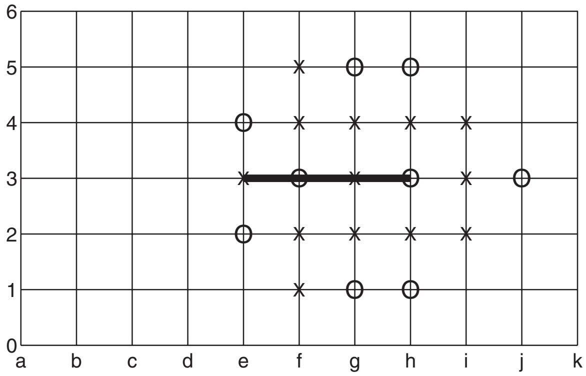


FIG. 8. *Step* 5.

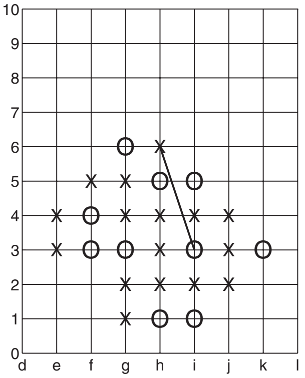


FIG. 9. *Step* 6.

Step 6. Add the edge marked in Figure 9, i.e., the edge $(i3, h6)$ ($h6 \in L_{\geq 3}$ since $I(g5) \neq I(h6)$).

LEMMA 7. *All the elements of $L_{\geq 3}$ have legal degrees in Γ_6.*

Proof.

- If $h7, i6 \in C$, then $h6 \in L_4$ and the possible neighbors of $h6$ (in addition to $i3$) are $f8, j8$ (Step 4), $h9$ (Step 5), and $i9, g9$ (Step 6). At most one of the vertices $f8$ and $g9$ belongs to C', and at most one of the vertices $h9$ and $i9$ belongs to C'; therefore the degree of $h6$ is at most 4.
- If $i6 \in C$ and $h7 \notin C$, then $h6 \in L_3$. The only neighbor of $h6$ is $i3$; therefore the degree of $h6$ is 1.
- If $h7 \in C$ and $i6 \notin C$, then $h6 \in L_3$. The possible neighbors of $h6$ (in addition to $i3$) are $f8, i9$, and at most one of them belongs to C', and hence the degree of $h6$ is at most 2. □

Step 7. Add the edge $(i3, f3)$ marked in Figure 10 if at least one of the vertices $e3, e4, f5$ belongs to C. Note that this edge should be added only once, even if $f2 \in C$ and at least one of the vertices $e2, e1, f0$ belongs to C.

LEMMA 8. *All the elements of $L_{\geq 3}$ have legal degrees in Γ_7.*

Proof.

- If $e3 \in C$, then $f3 \in L_{\geq 4}$. The possible neighbors of $f3$ (in addition to $i3$) are

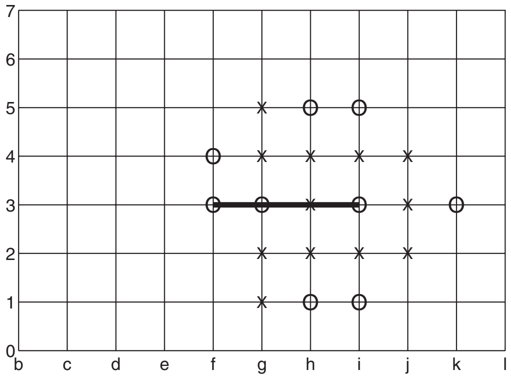


Fig. 10. *Step* 7.

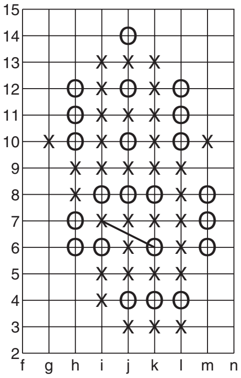


Fig. 11. *Step* 8.

$f1$ (Steps 2 and 3) and $c3$ (Step 7). Note that $f6 \notin C'$ because $I(g5) \neq I(h4)$, and for a similar reason $f0 \notin C'$. Also note that connections made by Steps 2 and 3 have two edges. Therefore the degree of $f3$ is at most 4.

- If $e3 \notin C$ and $f2 \in C$, then $f3 \in L_4$. The possible neighbors of $f3$ (in addition to $i3$) are $d2, d3, d4$ (Steps 2 and 3). It is possible for two of them to belong to C' only when $d2, d4 \in C'$, which is the case in Figure 6. Therefore the degree of $f3$ is at most 3.
- If $e3, f2 \notin C$ and $f5 \in C$, then $f3 \in \widetilde{L_3}$. The possible neighbors of $f3$ (in addition to $i3$) are $d3, d4, f1$ (Steps 2 and 3), at most two of them belong to C', and therefore the degree of $f3$ is at most 5.
- If $e3, f2, f5 \notin C$, then $e4 \in C$ (otherwise the edge is not added), and hence $f3 \in \widetilde{L_3}$. The only possible neighbor of $f3$ (in addition to $i3$) is $f1$ (Steps 2 and 3); therefore its degree is at most 3. □

Step 8. Add the marked edge in Figure 11, i.e., the edge $(k6, i7)$.

Lemma 9. *All the elements of $L_{\geq 3}$ have legal degrees in Γ_8.*

Proof. The only possible neighbor of $i7$ (in addition to $k6$) is $f6$ (Step 6); hence the degree of $i7$ is at most 2. □

Step 9. Add the edge $(k6, h6)$ marked in Figure 12 if $h6 \notin C$ or $h7 \notin C$.

Lemma 10. *All the elements of $L_{\geq 3}$ have legal degrees in Γ_9.*

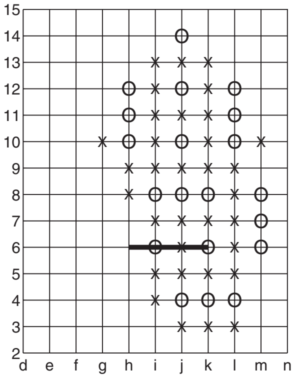


FIG. 12. *Step* 9.

Proof.

1. If $h6 \notin C$, then $h5 \in C$ (since $I(i5) \neq I(i6)$) and at least one of the vertices $g6, h7$ belongs to C (since $I(i5) \neq I(h6)$). Therefore $h6 \in L_{\geq 3}$.
   - If $g6, h7 \in C$, then $h6 \in L_4$. The possible neighbors of $h6$ (in addition to $k6$) are $e5, e7$ (Step 6), $f4$ (Step 4), and $e6$ (Steps 5 and 9). At most two of them belong to C'; therefore the degree of $h6$ is at most 3.
   - If $g6 \in C$ and $h7 \notin C$, then $g7 \in C$ (since $|I(h7)| > 0$) and the possible neighbors of $h6$ (in addition to $k6$) are $f4$ and $e7$, at most one of them belongs to C', and therefore the degree of $h6$ is at most 2.
   - If $h7 \in C$ and $g6 \notin C$, then $k6$ is the only neighbor of $h6$; therefore the degree of $h6$ is 1.
2. If $h6 \in C$, then $h7 \notin C$ (otherwise the edge is not added) and at least one of $g6, h5$ belongs to C (since $I(i6) \neq I(h6)$). Therefore $h6 \in L_{\geq 3}$.
   - If $h5 \in C$ and $g6 \notin C$, then the possible neighbors of $h6$ (in addition to $k6$) are $f5$ and $f6$ (Steps 2 and 3). At most one of them belongs to C'; therefore the degree of $h6$ is at most 2 if $h6 \in \overline{L_3}$ (Step 2 only) and at most 3 if $h6 \in \widetilde{L_3}$ (Steps 2 and 3).
   - If $h5, g6 \in C$, then the only possible neighbor of $h6$ (in addition to $k6$) is $e6$ (Steps 7 and 9); therefore its degree is at most 2.
   - If $h5 \notin C$, then $g6 \in C$. The only possible neighbor of $h6$ (in addition to $k6$) is $h4$ (Steps 2 and 3). It is impossible for $e6$ with Step 9 to be a neighbor of $h6$ because, in this case, $I(g11) = I(i11)$ or $I(h5) = I(h7)$ (it depends on the rotation/reflection of Figure 12 relative to $e6$). Therefore the degree of $h6$ is at most 2 if $h6 \in \overline{L_3}$ (Step 2) and at most 3 if $h6 \in \widetilde{L_3}$ (Steps 2 and 3). □

Step 10. For each edge (c_1, c_2), $c_1 \in C'$, $c_2 \in L_{\geq 3}$ in Γ_9, if the degree of c_2 is 1, then add an edge (c_1, c_2).

LEMMA 11. *All the elements of $L_{\geq 3}$ have legal degrees in Γ_{10}.*

Proof. For each vertex $c_2 \in L_{\geq 3}$, if the degree of c_2 in Γ_9 is 1, then its degree in Γ_{10} is 2; otherwise its degree does not change. □

Step 10 is the last step, and $\Gamma = \Gamma_{10}$. Lemma 11 states that the degrees of all the vertices of $L_{\geq 3}$ are legal in Γ, and it remains to prove the same about the elements of C'.

LEMMA 12. *In Figure* 5(c)*, the degree of $j6$ in Γ_{10} is at least* 4.

Proof. By Step 4, there are edges $(j6, h8)$ and $(j6, l8)$. Since $I(l6) \neq I(l7)$, $l6 \in L_{\geq 3}$ or $l7 \in L_{\geq 3}$. Therefore, by Step 2, there is in Γ_2 at least one of the edges $(j6, l6)$ and $(j6, l7)$. By the same argument, there is in Γ_2 at least one of the edges $(j6, h6)$ and $(j6, h7)$; hence the degree of $j6$ is at least 4. □

Lemma 13. *In Figure* 5(b)*, the degree of* $k6$ *in* Γ_{10} *is at least* 4.

Proof. If $m5 \in C$, then by Steps 2 and 3 there are two edges $(k6, m6)$ and two edges $(k6, m7)$, so assume that $m5 \notin C$. In this case, $l4 \in C$ since $|I(l5)| > 0$. We show that by Steps 3 and 10 there are two edges $(k6, m7)$: If $m7 \notin \overline{L_3}$, then there are two edges $(k6, m7)$ by Steps 2 and 3. If $m7 \in \overline{L_3}$, then $n6, n7, n8, m9 \notin C$. In this case, $o7 \in C$ (since $I(l7) \neq I(n7)$) and $o8 \in C$ (since $I(m9) \neq I(n8)$). It follows that the edge $(k6, m7)$ exists in Γ_2 since it is not omitted in a rotation of either Figure 5(a) or Figure 5(b). Steps 2, 7, and 9 leave the degree of $m7$ equal to one; hence if $m7 \in \overline{L_3}$, then the degree of $m7$ in Γ_9 is 1, and by Step 10 there are two edges $(k6, m7)$.

- If $j4 \notin C$, then $i5 \in C$ since $|I(j5)| > 0$. If neither of the edges $(k6, k4)$ and $(k6, l4)$ is in Γ_2, it can be only due to a rotation of Figure 5(b), in which case, by the same argument as for the edge $(k6, m7)$, there are two edges $(k6, i5)$, and we are done. If neither of the edges $(k6, i5)$ and $(k6, i6)$ are not in Γ_2, it can be only due to a rotation of Figure 5(b), but it is impossible since $I(h8) \neq I(i9)$. Therefore the degree of $k6$ is at least 4.
- If $j4 \in C$, then there is an edge $(k6, k4)$ and the nontrivial case is that in which $k4 \in \overline{L_3}$. If $i5 \in C$, then by Step 4 there is an edge $(k6, i4)$ and we are done, so assume that $i5 \notin C$. The situation is described in Figure 12, and the degree of $k6$ increases by either Step 8 or Step 9. □

Lemma 14. *In Figure* 5(a)*, the degree of* $k6$ *in* Γ_{10} *is at least* 4.

Proof. First note that $j4 \in C$ (since $|I(j5)| > 0$), $m7 \in C$ (since $|I(l7)| > 0$), and $m8 \in C$ (since $I(k9) \neq I(l8)$). If $m7 \in \overline{L_3}$ and $o7 \in C'$, then $l4 \in C$ (since $|I(l5)| > 0$) and $m4 \in C$ (since $I(m5) \neq I(n6)$), and there are two edges $(k6, l4)$ and two edges $(k6, k4)$. If $m7 \notin \overline{L_3}$ or $o7 \notin C'$, then by Step 3 or Step 10 there are two edges $(k6, m7)$. Also there are two edges $(k6, m6)$ or two edges $(k6, k4)$ (since $l4 \in C$ or $m5 \in C$ so that $|I(l5)| > 0$). Therefore the degree of $k6$ is at least 4. □

In what follows we assume that all the elements of C', for which we want to show that the degree is at least 4, have not been treated in Lemmas 12, 13, and 14. Lemmas 15 and 16 are taken from [6]. Lemma 15 is straightforward, and hence is given without a proof.

Lemma 15. *For every element of* C'*, which has not been treated in Lemmas* 12*,* 13*, and* 14*, the degree in* Γ_2 *is at least* 2.

Lemma 16. *If an element of* C' *has degree* 2 *in* Γ_2*, then its degree in* Γ *is at least* 4 *(again, an element which has not been treated in Lemmas* 12*,* 13*, and* 14*).*

Proof. An element of C' with degree 2 in Γ_2 must be of the form shown in Figure 13. If $b4 \notin C$, then $b3, b5 \in C$ (since $I(c3) \neq I(c4) \neq I(c5)$), and by Step 5 there is an edge $(e4, b4)$. If $b4 \in C$, then at least one of $b3$ and $b5$ belongs to C. Assume without loss of generality that $b5 \in C$. If $a4, a5, b6 \notin C$, then $c7 \in C$ (since $I(c6) \neq I(d5)$), and by Step 6 there is an edge $(e4, d7)$. If at least one of the vertices $a4, a5, b6$ belongs to C, then by Step 7 there is an edge $(e4, b4)$.

In a similar way, there is an edge that starts in $e4$ and is connected to $h4$ or $f1$ or $f7$; therefore the degree of $e4$ in Γ_7 is at least 4. □

Lemma 17. *If an element of* C' *has degree* 3 *in* Γ_2*, then its degree in* Γ *is at least* 4 *(again, an element which has not been treated in Lemmas* 12*,* 13*, and* 14*).*

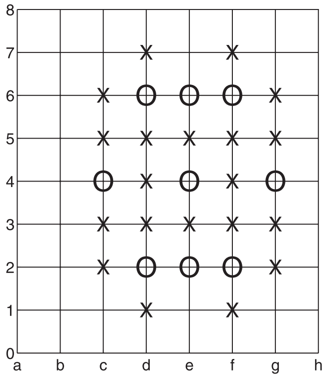


FIG. 13. *A codeword in* C' *with degree* 2 *in* Γ_2.

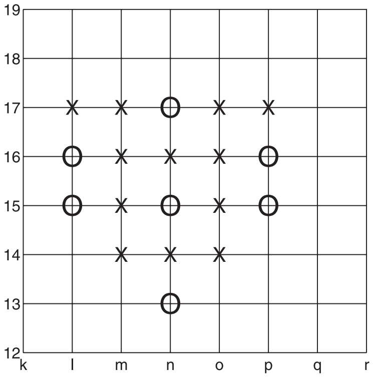


FIG. 14. *Illustration of case* 2(a) *of Lemma* 17.

Proof. Let $n15$ be a codeword in C' whose degree in Γ_2 is 3. Assume without loss of generality that $l16 \in C$. We distinguish between several cases as follows:

1. If $m17 \in C$, then $l17 \notin C$ (since the degree of $n15$ in Γ_2 is 3), and by Step 4 there is an edge $(n15, l17)$ in Γ.
2. If $m17 \notin C$:
   (a) If it is the case in Figure 14, the proof is similar to the proof of Lemma 16. If $n18 \notin C$ then $m18, o18 \in C$ and by Step 5 there is an edge $(n15, n18)$. If $n18 \in C$ then at least one of $m18$ and $o18$ belongs to C; without loss of generality assume that $o18 \in C$. If at least one of $n19, o19, p18$ belongs to C then by Step 7 there is an edge $(n15, n18)$. Otherwise $q17 \in C$ since $I(p17) \neq I(o16)$. If $q16 \notin C$ then by Step 6 there is an edge $(n15, q16)$. Otherwise $p16 \in \widetilde{L_3}$ and the degree of $n15$ in Γ_3 is at least 4.
   (b) If $l17 \in C$, then if $n15$ is the only neighbor of $l16$, then by Step 10 there are two edges $(n15, l16)$ and we are done. In the following subcases, $n15$ is not the only neighbor of $l16$.
      i. If $n17 \in C'$, then $p16 \in C$ (since $|I(o16)| > 0$). Since the degree of $n15$ in Γ_2 is 3, and $|I(m14)| > 0$ and $|I(o14)| > 0$, we get the configuration shown in Figure 15. If $n12 \in C$, then the degree of

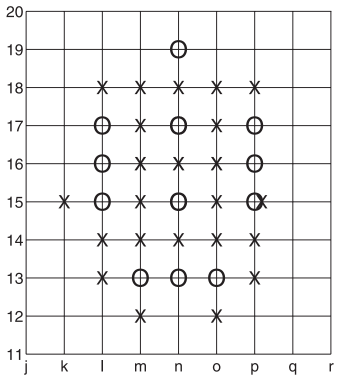


FIG. 15. *Illustration of Case* 2(b)i *of Lemma* 17.

$n15$ in Γ_3 is at least 4, and hence we assume that $n12 \notin C$. If $m11, n11, o11 \notin C'$, then $n15$ is the only neighbor of $n13$, and by Step 10 the degree of $n15$ in Γ is at least 4.

   A. If $n11 \in C'$, then $p12 \in C$ (since $I(o12) \neq I(o14)$), and $q13 \in C$ (since $I(o12) \neq I(p13)$). For similar reasons, $l12, k13 \in C$. The situation now is depicted in Figure 5(c), and again there are two edges $(n15, n13)$ by Step 10.

   B. If $o11 \in C'$, then $q12 \in C$ (since $|I(p12)| > 0$) and $q13 \in C$ (since $I(p13) \neq I(o14)$). This case is depicted in Figure 5(b). If in addition $m11 \in C'$, this is the case in Figure 5(a). Again, by Step 10, there are two edges $(n15, n13)$.

  ii. We assume now that $l16 \in \overline{L_3}$; otherwise there are two edges $(n15, l16)$ (Step 3) and we are done, unless it is the case in Figure 6, which has been treated in case 2(b)i. Therefore $m13 \in C$ (since $|I(m14)| > 0$). $j15, j17 \notin C'$ since $I(k16) \neq I(m16)$. Also, there is no edge $(l19, l16)$. The only possibility left, since $n15$ is not $l16$'s only neighbor, is that $j16 \in C'$. In this case, since $I(k15) \neq I(l14)$, $l13 \in C$. We repeat with $m13$ instead of $l16$, i.e., we have already treated all the cases except for the case in which $m13 \in \overline{L_3}$ and $m11 \in C'$. We have $p13, p14 \in C$ and repeat with $p14$ instead of $l16$, and finally $o17, p17 \in C$, and the degree of $n15$ is at least 4.

(c) If $l17 \notin C$, then since $I(l15) \neq I(l16)$, at least one of the vertices $l14, k15, k16$ belongs to C.

  i. If $l14 \in C$, the nontrivial case is that in which $l15 \in \overline{L_3}$. If $p16 \in C$, then we can assume that $o17 \notin C$ (otherwise it is case 1) and $p17 \notin C$ (otherwise it is case 2(b)). But now we are in case 2(a). Therefore we assume that $p16 \notin C$, and similarly we assume that $p14 \notin C$. Hence $o13, o17 \in C$ so that $|I(o14)| > 0$ and $|I(o16)| > 0$. We assume that $p13, p17 \notin C$ (otherwise it is case 2(b)), and we are again in case 2(a).

  ii. If $l14 \notin C$, then $m13 \in C$. We can assume that $l13 \notin C$ (otherwise it is case 2(b)) and $o13 \notin C$ (otherwise it is case 2(c)i). Therefore $p14 \in C$, and similarly $p13, p16 \notin C$. Hence $o17 \in C$ and the degree of $n15$ is at least 4. □

COROLLARY 4.1. *All the degrees in* Γ *are legal.*

As shown in section 3, this is enough to accomplish the proof of Theorem 1.

Acknowledgments. We would like to thank the anonymous referees.

REFERENCES

[1] U. BLASS, I. HONKALA, AND S. LITSYN, *On binary codes for identification*, J. Combin. Des., 8 (2000), pp. 151–156.

[2] U. BLASS, I. HONKALA, AND S. LITSYN, *Bounds on identifying codes*, Discrete Math., 241 (2001), pp. 119–128.

[3] I. CHARON, I. HONKALA, O. HUDRY, AND A. LOBSTEIN, *General bounds for identifying codes in some infinite regular graphs*, Electron. J. Combin., 8 (2001), Research Paper 39 http://www.combinatorics.org/Volume_8/v8i1toc.html.

[4] I. CHARON, O. HUDRY, AND A. LOBSTEIN, *Identifying codes with small radius in some infinite regular graphs*, Electron. J. Combin., 9 (2002), Research Paper 11 http://www.combinatorics.org/Volume_9/v9i1toc.html.

[5] G. COHEN, S. GRAVIER, I. HONKALA, A. LOBSTEIN, M. MOLLARD, C. PAYAN, AND G. ZÉMOR, *Improved identifying codes for the grid*, Electron. J. Combin., 6 (1999), Research Paper 19, Comment http://www.combinatorics.org/Volume_6/Html/v6i1r19.html.

[6] G. COHEN, I. HONKALA, A. LOBSTEIN, AND G. ZÉMOR, *New bounds for codes identifying vertices in graphs*, Electron. J. Combin. 6 (1999), Research Paper 19 http://www.combinatorics.org/Volume_6/v6i1toc.html.

[7] G. D. COHEN, I. HONKALA, A. LOBSTEIN, AND G. ZÉMOR, *Bounds for codes identifying vertices in the hexagonal grid* SIAM J. Discrete Math., 13 (2000), pp. 492–504.

[8] G. COHEN, I. HONKALA, A. LOBSTEIN, AND G. ZÉMOR, *On identifying codes*, in Codes and Association Schemes, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 56, AMS, New York, 2001, pp. 97–109.

[9] G. COHEN, I. HONKALA, A. LOBSTEIN, AND G. ZÉMOR, *On codes identifying vertices in the two-dimensional square lattice with diagonals*, IEEE Trans. Comput., 50 (2001), pp. 174–176.

[10] I. HONKALA AND T. LAIHONEN, *On the density of identifying codes in the square lattice*, J. Combin. Theory Ser. B, 85 (2002), pp. 297–306.

[11] I. HONKALA AND T. LAIHONEN, *On identifying codes in the hexagonal mesh*, Inform. Process. Lett., 89 (2004), pp. 9–14.

[12] I. HONKALA, T. LAIHONEN, AND S. RANTO, *On codes identifying sets of vertices in Hamming spaces*, Des. Codes Cryptogr., 24 (2001), pp. 193–204.

[13] I. HONKALA, T. LAIHONEN, AND S. RANTO, *On strongly identifying codes*, Discrete Math., 254 (2002), pp. 191–205.

[14] I. HONKALA AND A. LOBSTEIN, *On identifying codes in binary Hamming spaces*, J. Combin. Theory Ser. A, 99 (2002), pp. 232–243.

[15] I. HONKALA AND A. LOBSTEIN, *On the density of identifying codes in the square lattice*, J. Combin. Theory Ser. B, 85 (2002), pp. 297–306.

[16] M. KARPOVSKY, K. CHAKRABARTY, AND L. B. LEVITIN, *On a new class of codes for identifying vertices in graphs*, IEEE Trans. Inform. Theory, 44 (1998), pp. 599–611.

[17] T. LAIHONEN, *Optimal codes for strong identification* European J. Combin., 23 (2002), pp. 307–313.

[18] T. LAIHONEN, Sequences of optimal identifying codes, IEEE Trans. Inform. Theory, 48 (2002), pp. 774–776.

[19] T. LAIHONEN AND S. RANTO, *Families of optimal codes for strong identification*, Discrete Appl. Math., 121 (2002), pp. 203–213.

[20] A. LOBSTEIN, *A Two-Page Complement to "New Bounds for Codes Identifying Vertices in Graphs" by G. Cohen, I. Honkala, A. Lobstein, and G. Zémor, Published in Electron. J. Combin.,* 6 (1999)*, Research Paper* 19 *(electronic)*, manuscript, 2004. Available online at http://www.infres.enst.fr/∼lobstein/unpublished.html.

[21] S. RANTO, *Optimal linear identifying codes*, IEEE Trans. Inform. Theory, 49 (2003), pp. 1544–1547.

[22] S. RANTO, I. HONKALA, AND T. LAIHONEN, *Two families of optimal identifying codes in binary Hamming spaces*, IEEE Trans. Inform. Theory, 48(2002), no. 5, pp. 1200–1203.

SIAM J. DISCRETE MATH.
Vol. 19, No. 1, pp. 83–95

ON EQUITABLE COLORING OF d-DEGENERATE GRAPHS*

A. V. KOSTOCHKA[†], K. NAKPRASIT[‡], AND S. V. PEMMARAJU[§]

Abstract. An *equitable coloring* of a graph is a proper vertex coloring such that the sizes of any two color classes differ by at most 1. A *d-degenerate graph* is a graph G in which every subgraph has a vertex with degree at most d. A star S_m with m rays is an example of a 1-degenerate graph with maximum degree m that needs at least $1 + m/2$ colors for an equitable coloring. Our main result is that every n-vertex d-degenerate graph G with maximum degree at most $n/15$ can be equitably k-colored for each $k \geq 16d$. The proof of this bound is constructive. We extend the algorithm implied in the proof to an $O(d)$-factor approximation algorithm for equitable coloring of an *arbitrary* d-degenerate graph. Among the implications of this result is an $O(1)$-factor approximation algorithm for equitable coloring of planar graphs with fewest colors. A variation of equitable coloring (equitable partitions) is also discussed.

Key words. graph coloring, equitable coloring, d-degenerate graphs

AMS subject classifications. 05C15, 05C35, 05C85

DOI. 10.1137/S0895480103436505

1. Introduction.

An *equitable coloring* of a graph is a proper vertex coloring such that the sizes of every two color classes differ by at most 1. Equitable colorings naturally arise in some scheduling, partitioning, and load balancing problems [1, 2, 18, 23, 8, 24]. Pemmaraju [21] and Janson and Ruciński [11] used equitable colorings to derive deviation bounds for sums of dependent random variables that exhibit limited dependence. Subsequently, Janson [9] explored equitable colorings with applications to U-statistics, random strings, and random graphs. In these applications, the fewer colors we use, the better.

In contrast with ordinary coloring, a graph may have an equitable k-coloring (i.e., an equitable coloring with k colors) but no equitable $(k+1)$-coloring. It is easy to check that the complete bipartite graph $K_{7,7}$ has an equitable k-coloring for $k = 2, 4, 6$ and $k \geq 8$ but has no equitable k-coloring for $k = 3, 5, 7$. For a graph G, let $\mathrm{eq}(G)$ denote the smallest k_0 such that G is equitably k-colorable for every $k \geq k_0$.

Finding $\mathrm{eq}(G)$ even for planar graphs G is an NP-complete problem. In particular, determining if a given planar graph with maximum vertex degree 4 has an equitable coloring using at most 3 colors is NP-complete. This can be seen as follows. It is known [6] that determining if a planar graph with maximum vertex degree 4 is 3-colorable is NP-complete. For a given n-vertex planar graph G with maximum vertex degree 4, let G' be obtained from G by adding $2n$ isolated vertices. Then G is 3-colorable if and only if G' is equitably 3-colorable.

This NP-completeness result motivates a series of extremal problems on equitable

*Received by the editors October 23, 2003; accepted for publication (in revised form) November 4, 2004; published electronically June 16, 2005.
http://www.siam.org/journals/sidma/19-1/43650.html

[†] Department of Mathematics, The University of Illinois, Urbana, IL 61801, and Institute of Mathematics, Novosibirsk, Russia (kostochk@math.uiuc.edu). This author's work was partially supported by NSF grant DMS-0099608 and by grants 02-01-00039 and 00-01-00916 of the Russian Foundation for Basic Research.

[‡] Department of Mathematics, The University of Illinois, Urbana, IL 61801 (nakprasi@math.uiuc.edu). This author's work was partially supported by NSF grant DMS-0099608.

[§] Department of Computer Science, The University of Iowa, Iowa City, IA 52242 (sriram@cs.uiowa.edu). This author's work was partially supported by NSF grant DMS-0213305.

colorings. A typical problem would ask us to show that if a graph G is “sparse,” then $\mathrm{eq}(G)$ is “small.” Here “sparse” might mean that G has a small maximum degree, or small average degree, or is d-degenerate for a small d. Recall that a graph G is *d-degenerate* if every subgraph G' of G has a vertex with degree (in G') at most d. It is well known that forests are exactly 1-degenerate graphs, outerplanar graphs are 2-degenerate, and planar graphs are 5-degenerate. By definition, the vertices of every d-degenerate graph can be ordered $v_1, \dots, v_n$ in such a way that for every $i \geq 2$, vertex v_i has at most d neighbors v_j with $j < i$.

Hajnal and Szemerédi [7] considered the first version of “sparseness” of a graph. They settled a conjecture of Erdős by proving that every graph G with maximum degree at most Δ has an equitable k-coloring for every $k \geq 1 + \Delta$. In other words, they proved that $\mathrm{eq}(G) \leq \Delta(G) + 1$ for every graph G. In its “complementary” form, this result concerns the decomposition of a sufficiently dense graph into cliques of equal size, which has been used in a number of applications of Szemerédi’s regularity lemma [13]. The bound of the Hajnal–Szemerédi theorem is sharp, but it can be improved for some important classes of graphs. In fact, Chen, Lih, and Wu [5] conjectured that every connected graph G with maximum degree $\Delta \geq 2$ has an equitable coloring with Δ colors, except when G is a complete graph or an odd cycle or Δ is odd and $G = K_{\Delta,\Delta}$. They proved the conjecture for graphs with maximum degree at most 3. Lih and Wu [19] proved the conjecture for bipartite graphs and Yap and Zhang [25, 26] proved that the conjecture holds for outerplanar graphs and planar graphs with maximum degree at least 13. In an unpublished paper, Nakprasit extended the result of Yap and Zhang [26] to planar graphs with maximum degree at least 9.

If a graph G has moderate maximum degree Δ and, in addition, is d-degenerate for a small d, then one can get a somewhat better than Δ bound on $\mathrm{eq}(G)$. Meyer [20] proved that every forest (i.e., 1-degenerate graph) with maximum degree Δ has an equitable coloring with $1 + \lceil \Delta/2 \rceil$ colors. This bound is attained at the star S_m with m rays: in every proper coloring of S_m, the center vertex forms a color class, and hence the remaining vertices need at least $m/2$ colors. Kostochka and Nakprasit [15] obtained the upper bound $\mathrm{eq}(G) \leq (d + \Delta + 1)/2$ for d-degenerate graphs with maximum degree Δ in the case $\Delta \geq 27d$. This bound is also sharp.

Bollobás and Guy [4] initiated a new and important direction of research for equitable colorings. They showed that while $1 + \lceil \Delta/2 \rceil$ is a tight upper bound on the equitable chromatic number of trees, “most” trees can be equitably 3-colored. Their result implies that each n-vertex forest F with $\Delta(F) \leq n/3$ can be equitably 3-colored. This result seems to uncover a fundamental phenomenon in equitable colorings: apart from some “star-like” graphs, most graphs admit equitable colorings with few colors. Another example of this phenomenon was given by Pemmaraju [22]. He showed that every n-vertex outerplanar graph G with $\Delta(G) \leq n/6$ can be equitably 6-colored. In this paper we show that this phenomenon is widely pervasive.

Our main result is the following.

THEOREM 1. *For $d, n \geq 1$, every d-degenerate, n-vertex graph G with $\Delta \leq n/15$ is equitably k-colorable for each $k \geq 16d$.*

The proof of Theorem 1 is constructive and provides an $O(d)$-factor approximation algorithm for equitable coloring with fewest colors of each d-degenerate n-vertex graph G with $\Delta \leq n/15$. Furthermore, many d-degenerate graphs need at least $\Omega(d)$ colors for ordinary coloring, and for such graphs our algorithm gives a constant factor (independent of d) approximation. Then we extend the algorithmic side of Theorem 1 to *all* d-degenerate graphs and show the following.

THEOREM 2. *There exists a polynomial time algorithm that for every equitably s-colorable d-degenerate graph G produces an equitable k-coloring of G for any $k \geq 31ds$.*

The result of Theorem 2 was already used by Bodlaender and Fomin [3] for constructing a polynomial time algorithm for equitable coloring of graphs with a bounded tree width. Theorem 2 gives an $O(d)$-factor approximation algorithm for the problem of the equitable coloring of a d-degenerate graph with fewest colors. For some classes of graphs such as planar graphs, this translates into an $O(1)$-factor approximation algorithm.

The technique used for the proof of Theorem 1 allows us to treat the following variation of equitable coloring. An *equitable k-partition* of a graph G is a collection of subgraphs $\{G[V_1], G[V_2], \dots, G[V_k]\}$ of G induced by the vertex partition $\{V_1, V_2, \dots, V_k\}$ of $V(G)$ where, for every pair V_i and V_j, the sizes of V_i and V_j differ by at most 1. Certainly, every equitable coloring is an equitable partition. Pemmaraju [22] proved that every outerplanar graph has an equitable partition into two forests.

THEOREM 3. *Let $k \geq 3$ and $d \geq 2$. Then every d-degenerate graph has an equitable k-partition into $(d-1)$-degenerate graphs.*

This is an extension of the Bollobás–Guy result [4], which essentially asserts the same for $d = 1$ and $k = 3$. Note that there is no restriction on the maximum degree of a graph in Theorem 3, while such a restriction is important in the Bollobás–Guy theorem.

2. Coloring d-degenerate graphs with $O(d)$ colors.

An enumeration $v_1, v_2, \dots, v_n$ of the vertices of a graph G is a *greedy enumeration* (or a *greedy order*) if for every i, $1 \leq i \leq n$, the vertex v_i is a vertex of maximum degree in $G - v_1 - \cdots - v_{i-1}$. Similarly, the enumeration or order is *degenerate* if for every i, $1 \leq i \leq n$, the vertex v_i has minimal degree in $G(\{v_1, \dots, v_i\})$. Note that if $v_1, v_2, \dots, v_n$ is a greedy order on G, then $v_i, v_{i+1}, \dots, v_n$ is a greedy order on $G - v_1 - \cdots - v_{i-1}$, and that if $v_1, v_2, \dots, v_n$ is a degenerate order on G, then $v_1, v_2, \dots, v_i$ is a degenerate order on $G - v_{i+1} - \cdots - v_n$.

If G is d-degenerate, then, by the very definition, in every degenerate order $v_1, v_2, \dots, v_n$ of G, every v_i has at most d neighbors v_j with $j < i$.

The main result of section 2 is Theorem 1 whose statement we repeat below for the reader's convenience.

THEOREM 4 (restatement of Theorem 1). *Every d-degenerate graph with maximum degree at most Δ is equitably k-colorable when $k \geq 16d$ and $n \geq 15\Delta$.*

Proof. Let G be a d-degenerate graph with vertex set V of size n and edge set $E(G)$. Let t be an integer such that $k(t-1) < n \leq kt$ and $k \geq 16d$.

Case 1. $t \leq 15$. We will color the vertices one by one in a degenerate order $v_1, \dots, v_n$ (with some recolorings). Suppose we cannot color vertex v_i. Let Z be the set of color classes containing neighbors of v_i. Since G is d-degenerate, $|Z| \leq d$. If a color class $M \notin Z$ has fewer than t vertices, then we can color v_i with M. Since $n \leq kt$, there is a color class $M_0 \in Z$ with at most $t-1$ vertices. If a vertex w in a color class $M \notin Z$ has no neighbors in M_0, then we can recolor w with M_0 and color v_i with M. Thus, each of the $(k - |Z|)t$ colored vertices outside of Z has a neighbor in M_0. Therefore,

$$(t-1)\Delta \geq (k-d)t\frac{15}{16}kt \geq \frac{15}{16}n.$$

Since $n \geq 15\Delta$, we have

$$(t-1)\frac{n}{15} \geq \frac{15}{16}n,$$

and hence $t-1 \geq 15^2/16 > 14$, which contradicts the choice of t.

Case 2. $t \geq 16$. Let $t = \beta_1 4^m + \beta_2 4^{m-1} + \cdots + \beta_{m+1}$, where β_j is an integer, $0 \leq \beta_j \leq 3$. For $i = 1, 2, \ldots, m+1$, define $l_i = \beta_1 4^{i-1} + \beta_2 4^{i-2} + \cdots + \beta_i$. For notational convenience, let $l_0 = 0$. We have that $l_i = 4l_{i-1} + \beta_i$ for each $i = 1, 2, \ldots, m+1$ and also that $t = l_{m+1}$.

We now partition V into sets $C_1, C_2, \ldots, C_{m+1}$ and color the vertices in C_i at the ith phase of the algorithm. We use the values of $l_1, l_2, \ldots, l_m$ to control the sizes of these sets. For convenience, set $A_0 = B_0 = C_0 = \emptyset$. For each $i = 1, 2, \ldots, m$, we construct sets A_i and B_i and let $C_i = A_i \cup B_i$. We use C'_i to denote the vertices in the sets constructed thus far. In other words, for each $i = 0, 1, \ldots, m+1$, we let C'_i denote $\cup_{j=0}^{i} C_j$. For each $i = 1, 2, \ldots, m$, A_i is constructed by selecting vertices in $G - C'_{i-1}$ as follows. Arrange the vertices of $G - C'_{i-1}$ in a greedy ordering and let A_i be the first $(l_i - l_{i-1})k$ vertices in this ordering. B_i is selected from vertices in $G - C'_{i-1} - A_i$ as follows. Initially set $B_i = \emptyset$ and, while there is a vertex $w \in G - C'_{i-1} - A_i - B_i$ that has at least $13d$ neighbors in $A_i \cup B_i \cup C'_{i-1}$, add w to B_i. Repeat this process until every vertex $w \in G - C'_{i-1} - A_i - B_i$ has fewer than $13d$ neighbors in $C'_{i-1} \cup A_i \cup B_i$. This completes the construction of A_i and B_i and we simply set $C_i = A_i \cup B_i$. After constructing $C_1, C_2, \ldots, C_m$, we set $C_{m+1} = V(G) - C'_m$.

Now let $b_i = |B_i|$ for each $i = 0, 1, 2, \ldots, m$ and let $e(H)$ denote the number of edges in a graph H. It follows from our construction that for each $i = 0, 1, \ldots, m$,

$$e(G[C'_i]) \geq 13d \sum_{j=0}^{i} b_j.$$

On the other hand, $G[C'_i]$ is a d-degenerate graph and has $l_i k + \sum_{j=0}^{i} b_j$ vertices, and so $e(G[C'_i]) < (l_i k + \sum_{j=0}^{i} b_j)d$. It follows that $\sum_{j=0}^{i} b_j < (l_i k/12)$, or in other words, for each $i = 1, \ldots, m$,

$$|C'_i| < \frac{13}{12} l_i k. \tag{1}$$

Since $C'_{m+1} = V(G)$, we also know that $|C'_{m+1}| \leq tk = l_{m+1}k$.

We will color C_1 with k colors in such a way that each color class has at most $\lceil \frac{7}{6} l_1 \rceil$ vertices. We color vertices in C_1 one by one in a degenerate order. Hence when we color vertex $u \in C_1$, there are at least $k-d$ color classes that do not contain neighbors of u. Since

$$|C_1| < \frac{13 l_1 k}{12} \leq \frac{13 l_1 k}{12} \frac{16(k-d)}{15k} < \frac{7}{6} l_1 (k-d),$$

there exists a color class M of size less than $\frac{7}{6} l_1$ that does not contain neighbors of u. We color u with color M.

We now show how to color the rest of the sets $C_2, C_3, \ldots, C_{m+1}$. For $2 \leq i \leq m+1$, at the ith phase we start with G such that all vertices in C'_{i-1} have been colored. At this phase we will color the vertices in C_i in a degenerate order in such a way that (i) every color class is of size at most L_i, where $L_i = \lceil \frac{7}{6} l_i \rceil$ for $2 \leq i \leq m$, and $L_{m+1} = t$; (ii) the vertices in C'_{i-1} will *not* be recolored.

CLAIM 2.1. *For every $i \geq 2$, $L_{i-1}/L_i \leq 2/5$.*

Proof. Recall that $l_i \geq 4l_{i-1}$ for every $i \geq 2$. If $i = m+1$, then $L_i = l_i = t \geq 16$. Therefore,

$$\frac{L_m}{L_{m+1}} = \frac{\lceil 7l_m/6 \rceil}{t} \leq \frac{7l_m/6 + 5/6}{t} \leq \frac{7}{6 \cdot 4} + \frac{5/6}{16} = \frac{11}{32} < \frac{2}{5}.$$

If $2 \leq i \leq m$, then $L_i = \lceil \frac{7l_i}{6} \rceil$. If $l_{i-1} \geq 2$, then $l_i \geq 8$ and

$$\frac{L_{i-1}}{L_i} \leq \frac{7l_{i-1}/6 + 5/6}{7l_i/6} \leq \frac{1}{4} + \frac{5/6}{7 \cdot 8/6} = \frac{19}{56} < \frac{2}{5}.$$

Finally, if $l_{i-1} = 1$, then $L_{i-1} = 2$ and $L_i \geq 5$. This proves the claim. □

Suppose we want to color a vertex v. Let $M_1, \ldots, M_k$ be the current color classes. Let Y_0 denote the set of color classes of cardinality less than L_i. If some $M_j \in Y_0$ contains no neighbors of v, then we color v with M_j and work with the next vertex. Otherwise, let Y_0-*candidate* be a vertex $w \in V - C'_{i-1}$ such that there exists a color class $M(w) \in Y_0$, with $w \notin M(w)$ and $N_G(w) \cap M(w) = \emptyset$. Let Y_1 be the set of color classes containing a Y_0-candidate. If a member M_j of Y_1 does not contain a neighbor of v, then we color v with M_j and recolor some Y_0-candidate $w \in M_j$ with $M(w)$. For $h \geq 1$, let a Y_h-*candidate* be a vertex $w \in C_i - \cup_{M \in Y_0 \cup \cdots \cup Y_h} M$ such that there exists $M(w) \in Y_h$ with $N_G(w) \cap M(w) = \emptyset$. Let Y_{h+1} be the set of color classes containing a Y_h-candidate. If a member M_j of Y_{h+1} does not contain a neighbor of v, then we color v with M_j and similarly to the above recolor a sequence of candidates. Finally, let $Y = \cup_{j=0}^{\infty} Y_j$ and $y = |Y|$. Then by the above, Y possesses the following properties:

(a) Every color class in Y contains a neighbor of v.

(b) Every vertex $u \in C_i - \cup_{M \in Y} M$ has a neighbor in every $M \in Y$ (otherwise the color class of u would be in Y).

We will prove that there is at least one color class M in Y that does not contain neighbors of v. Suppose this is not the case.

Observe that each vertex $u \in C_i$ has less than $13d$ neighbors in C'_{i-1} (by the construction of B_{i-1}) and at the moment of coloring has at most d neighbors among vertices of C_i colored earlier (since vertices are considered in a degenerate order). So when we color a vertex $u \in C_i$, there are less than $(13+1)d$ color classes that have neighbors of u. By property (a) of Y, $y < 14d$.

CLAIM 2.2. $y < 8d/7$.

Proof. Let $S = \cup_{M \in Y} M$ and $T = C_i - S$. By property (b) of Y, at least $y|T|$ edges connect T with S. Since G is d-degenerate, we conclude that $y|T| < d(|S| + |T|)$, i.e., that $(y-d)|T| < d|S|$. Clearly, $|S| \leq yL_i$. By the definition of Y_0, every color class outside of Y_0 has size exactly L_i, and each $k - y$ color class outside of Y contains at most L_{i-1} vertices in C'_{i-1}. Hence

$$|T| \geq (k-y)(L_i - L_{i-1}).$$

By Claim 2.1, $\frac{L_i - L_{i-1}}{L_i} \geq 1 - \frac{2}{5} = \frac{3}{5}$ for every $i \geq 2$. Therefore,

$$(y-d)(k-y)\frac{3}{5} < dy.$$

Since $k \geq 16d$, the last inequality yields that $(y-d)(16d-y)\frac{3}{5} < dy$. This implies the following inequality for $\gamma = y/d$:

$$\gamma^2 - \frac{46}{3}\gamma + 16 > 0.$$

Therefore, either $\gamma > (23+\sqrt{385})/3 \sim 14.207\ldots$ or $\gamma < (23-\sqrt{385})/3 \sim 1.1261\ldots < 8/7$. The former is impossible since $y \leq 14d$, and thus the latter holds. This proves the claim. □

Subcase 2.1. $2 \leq i \leq m$. The total number of colored vertices is at least $L_i(k-y)$, which by Claim 2.2 is greater than

$$\left\lceil \frac{7l_i}{6} \right\rceil \left(k - \frac{8d}{7} \right) \geq \frac{7l_i}{6}\frac{13k}{14} = \frac{13l_ik}{12}.$$

This contradicts (1) for $j = i-1$.

Subcase 2.2. $i = m+1$. Let D_i be the highest degree in $G[V - C_i']$.

Claim 2.3. $l_1\Delta + (l_2-l_1)D_1 + (l_3-l_2)D_2 + \cdots + (l_{m+1}-l_m)D_m \leq 3\Delta + 4.25dt$.

Proof. Observe that

$$|E(G)| \geq \sum_{\substack{1\leq i\leq m\\ 1\leq j\leq l_ik}} \deg_{V-C'_{i-1}-\{v_1^i,\ldots,v_{j-1}^i\}}(v_j^i)\ldots.$$

By the definition of A_i, for $v_j^i \in A_i$,

$$\deg_{G[V-C'_{i-1}-\{v_1^i,\ldots,v_{j-1}^i\}]}(v_j^i) \geq D_i \quad \text{and} \quad |A_i| = (l_i - l_{i-1})k.$$

Thus,

$$|E(G)| \geq k(l_1D_1 + (l_2-l_1)D_2 + (l_3-l_2)D_3 + \cdots + (l_m - l_{m-1})D_m).$$

Since $|E(G)| < dn \leq dtk$, we have

$$l_1D_1 + (l_2-l_1)D_2 + (l_3-l_2)D_3 + \cdots + (l_m-l_{m-1})D_m < dt. \tag{2}$$

Note that

$$\frac{l_{i+1}-l_i}{l_i-l_{i-1}} = \frac{4l_i + \beta_{i+1} - l_i}{4l_{i-1}+\beta_i - l_{i-1}} \leq \frac{3(4l_{i-1}+\beta_i)+3}{3l_{i-1}+\beta_i} = 4 + \frac{3-\beta_i}{3l_{i-1}+\beta_i} \leq 4 + \frac{1}{l_{i-1}}.$$

For $i \geq 3$, we obtain $l_{i+1} - l_i \leq (4+\frac{1}{4})(l_i - l_{i-1})$. Also $(l_2 - l_1) - 4.25l_1 = \beta_2 - 1.25l_1$. Therefore,

$$\begin{aligned} &4.25\,(l_1D_1 + (l_2-l_1)D_2 + (l_3-l_2)D_3 + \ldots + (l_m-l_{m-1})D_m) \\ &\quad \geq (l_2-l_1)D_1 + (l_3-l_2)D_2 + \cdots + (l_{m+1}-l_m)D_m + (1.25l_1 - \beta_2)D_1. \end{aligned}$$

Comparing with (2), we get

$$(l_2-l_1)D_1 + (l_3-l_2)D_2 + \cdots + (l_{m+1}-l_m)D_m < 4.25dt + \beta_2D_1 - 1.25l_1D_1.$$

Hence

$$l_1\Delta + (l_2-l_1)D_1 + (l_3-l_2)D_2 + \cdots + (l_{m+1}-l_m)D_m \leq l_1\Delta + 4.25dt + \beta_2D_1 - \frac{5}{4}l_1D_1.$$

In order to prove the claim it is now enough to show that

$$l_1\Delta + \beta_2D_1 - \frac{5}{4}l_1D_1 \leq 3\Delta. \tag{3}$$

Recall that $l_1 \leq 3$ and $\beta_2 \leq 3$. If $\beta_2 \leq \frac{5}{4}l_1$, then (3) is evident. If $\beta_2 > \frac{5}{4}l_1$, then

$$l_1\Delta + \beta_2 D_1 - \frac{5}{4}l_1 D_1 \leq l_1\Delta + \left(\beta_2 - \frac{5}{4}l_1\right)\Delta \leq \beta_2\Delta \leq 3\Delta.$$

This proves (3) and thus the claim. □

Let $M_1 \in Y_0$. By construction, every M_j contains at most L_i vertices in C'_i. So the number of neighbors of M_1 is at most

$$\begin{aligned}
&L_1\Delta + (L_2 - L_1)D_1 + \cdots + (L_{m+1} - L_m)D_m \\
&= \left\lceil \frac{7l_1}{6} \right\rceil \Delta + \left(\left\lceil \frac{7l_2}{6} \right\rceil - \left\lceil \frac{7l_1}{6} \right\rceil \right) D_1 + \cdots + \left(t - \left\lceil \frac{7l_m}{6} \right\rceil \right) D_m \\
&= \left\lceil \frac{7l_1}{6} \right\rceil (\Delta - D_1) + \left\lceil \frac{7l_2}{6} \right\rceil (D_1 - D_2) + \cdots + \left\lceil \frac{7l_m}{6} \right\rceil (D_{m-1} - D_m) + tD_m \\
&\leq \frac{7l_1}{6}(\Delta - D_1) + \frac{5}{6}(\Delta - D_1) + \frac{7l_2}{6}(D_1 - D_2) + \frac{5}{6}(D_1 - D_2) \\
&\quad + \cdots + \frac{7l_m}{6}(D_{m-1} - D_m) + \frac{5}{6}(D_{m-1} - D_m) + tD_m \\
&\leq \left(\frac{7l_1}{6} + \frac{5}{6} \right)\Delta + \frac{7}{6}\left((l_2 - l_1)D_1 + (l_3 - l_2)D_2 + \cdots + (l_{m+1} - l_m)D_m \right).
\end{aligned}$$

On the other hand, as in the proof of Claim 2.2, every color class outside of Y_0 has size exactly $L_{m+1} = t$, and each of the $k-y$ color classes outside of Y contains at most L_m vertices in C'_m. Hence, the number of neighbors of M_1 is at least $(k-y)(t-L_m)$. Note that

$$t - L_m = t - \left\lceil \frac{7l_m}{6} \right\rceil \geq t\left(1 - \frac{\frac{7l_m}{6} + \frac{5}{6}}{t}\right) = t\left(1 - \frac{7}{4\cdot 6} - \frac{5}{6 \cdot 16}\right) = \frac{21}{32}t.$$

Hence by Claim 2.3 we have

$$(k-y)(t-L_m) \geq \left(k - \frac{8d}{7}\right)\frac{21}{32}t.$$

Comparing this with the upper bound above and applying Claim 2.3 we get

$$\left(k - \frac{8d}{7}\right)\frac{21}{32}t \leq \frac{5}{6}\Delta + \frac{7}{6}\left(3\Delta + 4.25dt\right).$$

Since $\Delta \leq n/15 \leq kt/15$, this reduces to

$$\left(k - \frac{8d}{7}\right)\frac{21}{32} \leq \frac{5}{6 \cdot 15}k + \frac{7}{6}\left(\frac{3}{15}k + 4.25d\right),$$

which gives

$$\left(\frac{21}{32} - \frac{1}{18} - \frac{7}{6}\frac{1}{5}\right)k \leq \left(\frac{21}{32}\frac{8}{7} + \frac{7 \cdot 4.25}{6}\right)d.$$

It follows that

$$\frac{k}{d} \leq \frac{68.5}{12}\frac{1440}{529} = \frac{8220}{529} < 15.6,$$

which contradicts $k \geq 16d$. This proves the theorem. □

Algorithm. The above proof implies a simple algorithm for equitable k-coloring of any n-vertex d-degenerate graph with $\Delta(G) \le n/15$. We first partition $V(G)$ into sets C_i, $1 \le i \le m+1$, as described in the first part of the proof. Then for each $i = 1, 2, \dots, m+1$, we attempt to color vertices of C_i in degenerate order. It is possible that in the process some vertices may have to be recolored, but these recolorings are restricted to the set currently being colored, namely, C_i. The algorithm clearly runs in polynomial time and it can be implemented in $O(n^3)$ time; we do not give details here.

3. Constant-factor approximation algorithm. The algorithm above can be thought of as providing an $O(d)$-factor approximation algorithm for equitable coloring with fewest colors of an n-vertex d-degenerate graph with maximum degree at most $n/15$. In this section, we extend this to an $O(d)$-factor algorithm for equitable coloring of an *arbitrary* d-degenerate graph. This implies an $O(1)$-factor algorithm for planar graphs. The main result in this section is the following.

Theorem 5. *Every n-vertex d-degenerate graph G with maximum degree at most Δ is equitably k-colorable for any k, $k \ge \max\{62d,\, 31d\frac{n}{n-\Delta+1}\}$.*

Proof. Let G be an n-vertex d-degenerate graph. Let $G_0 = G$, $h = 30d-1$ and, for $j = 1, \dots, h$, let w_j be a vertex of the maximum degree in G_{j-1} and $G_j = G_{j-1} - w_j$.

Claim 3.1. *For every $v \in V(G_h)$, $\deg_{G_h}(v) < n/30$.*

Proof. If $\deg_{G_h}(v) \ge n/30$ for some $v \in V(G_h)$, then also $\deg_{G_{j-1}}(w_j) \ge n/30$ for every $j = 1, \dots, 30d-1$, and hence $|E(G)| \ge 30d(n/30) = dn$. This is a contradiction, since any n-vertex d-degenerate graph has fewer than dn edges. □

Claim 3.2. *There are pairwise disjoint independent sets $M_1, M_2, \dots, M_h$ such that for every j, $1 \le j \le h$,*

(i) $w_j \in \bigcup_{s=1}^{j} M_s$,

(ii) $\lfloor n/k \rfloor \le |M_j| \le \lceil n/k \rceil$, *and*

(iii) $nj/k \le \sum_{s=1}^{j} |M_s| < 1 + nj/k$.

Proof. Let $X_1 = V(G) - w_1 - N_G(w_1)$. Clearly, $|X_1| \ge n - \Delta - 1$. Since G is d-degenerate, X_1 contains an independent set M_1' of size at least $\frac{|X_1|}{d+1} \ge \frac{n-\Delta-1}{d+1}$. Since

$$\frac{n}{k} \le \frac{n-\Delta+1}{31d} < \frac{n-\Delta}{d+1},$$

$|M_1'| > \frac{n}{k} - \frac{1}{d+1}$. Hence, we can choose a subset M_1'' of M_1' of size $\lceil \frac{n}{k} \rceil - 1$ and let $M_1 = M_1'' + w_1$. By construction, M_1 satisfies properties (i)–(iii) for $j = 1$.

Suppose we have constructed $M_1, M_2, \dots, M_{j-1}$ satisfying (i)–(iii) for some $j \le h$. Let $x_j = w_j$ if $w_j \notin \bigcup_{s=1}^{j-1} M_s$, and let x_j be any vertex outside $\bigcup_{s=1}^{j-1} M_s$ otherwise. Let $X_j = V(G) - \bigcup_{s=1}^{j-1} M_s - x_j - N_G(x_j)$. Since G is d-degenerate, X_j contains an independent set M_j' of size at least $\frac{|X_j|}{d+1}$. Suppose that $|M_j'| < -1 + n/k$. In view of (iii), this means that

$$\frac{n - 1 - (j-1)\frac{n}{k} - 1 - \Delta}{d+1} < \frac{n}{k} - 1.$$

For $n > k$ and $d \ge 1$, the last inequality yields $n - \Delta + 1 < \frac{(j+d)n}{k} + 1 < \frac{31dn}{k}$. But this contradicts the choice of k. Thus, we can choose a subset of M_j' that together with x_j forms an independent set M_j'' of size $\lceil n/k \rceil$. If

$$|M_j''| + \sum_{s=1}^{j-1} |M_s| < \frac{jn}{k} + 1,$$

then we let $M_j = M_j''$; otherwise we get M_j by deleting a vertex $v \neq x_j$ from M_j''. Note that in the latter case, $\lfloor n/k \rfloor \neq \lceil n/k \rceil$, and thus (i)–(iii) hold in both cases. This proves the claim. $\square$

Let G' be the graph obtained by deleting vertices in $M_1 \cup M_2 \cup \cdots \cup M_h$ from G and let $V' = V(G')$.

CLAIM 3.3. $|V'| \geq 16n/31$.

Proof. By (iii) of Claim 3.2, $|V'| \geq n - (30d-1)n/k - 1 \geq n - 30dn/k$. Since $k \geq 62d$, we get $|V'| \geq 32n/62$. $\square$

By Claims 3.1 and 3.3,

$$\frac{|V'|}{\Delta(G')} \geq \frac{32n}{62} \cdot \frac{30}{n} > 15.$$

Since $k-h \geq 62d - 30d = 32d$, by Theorem 1, G' is equitably $(k-h)$-colorable. Hence G is equitably k-colorable. This proves the theorem. $\square$

COROLLARY 1. *Every d-degenerate graph with n vertices and maximum degree at most $1 + n/2$ is equitably k-colorable when $k \geq 62d$.*

Now we are ready to prove Theorem 2, which we state again for convenience.

THEOREM 6 (restatement of Theorem 2). *There exists a polynomial time algorithm that, given a d-degenerate graph G with $\chi_{eq}(G) \leq s$, can equitably color G with k colors for any k, $k \geq 31ds$.*

Proof. Assume that a graph G on n vertices with maximum degree Δ admits an equitable coloring ϕ with s colors. Let $v \in V(G)$ have degree Δ. The color class of v contains at most $n - \Delta$ vertices. Thus no other color class can contain more than $n - \Delta + 1$ vertices. Hence,

$$s > \frac{n}{n - \Delta + 1}. \tag{4}$$

Also, if G has at least one edge, $s \geq 2$. If $\Delta \leq 1 + n/2$, then by Corollary 1 G can be equitably k-colored for any $k \geq 62d$. Since $62d \leq 31ds$, G can be equitably k-colored for any $k \geq 31ds$. If $\Delta > 1 + n/2$, then $31d\frac{n}{n-\Delta+1} > 62d$ and therefore by Theorem 5, G can be equitably k-colored for any $k \geq 31d\frac{n}{n-\Delta+1}$. It follows from inequality (4) that G can be equitably k-colored for any $k \geq 31ds$.

The fact that such an equitable k-coloring can be constructed in polynomial time is implied by the proof of Theorem 5. The algorithm is sketched here. First identify the high degree vertices $w_1, w_2, \ldots, w_h$ in G and construct the color classes $M_1, M_2, \ldots, M_h$ containing these vertices as in Claim 3.1. Construction of these color classes uses as a subroutine an algorithm that finds an independent set of size at least $m/(d+1)$ in a given m-vertex, d-degenerate graph. The following greedy algorithm suffices for this task: pick a minimum degree vertex, delete the vertex and its neighbors, and repeat until no vertices are left. Since at every step we deleted at most $d+1$ vertices, the number of steps will be at least $m/(d+1)$. Once the color classes $M_1, M_2, \ldots, M_h$ are constructed and the colored vertices are deleted, we are left with a graph whose maximum vertex degree is less than $n/30$. We color the vertices in this graph using the algorithm from the previous section. This phase dominates the running time of the algorithm, and hence we have an $O(n^3)$ algorithm. $\square$

4. Equitable partitions of *d*-degenerate graphs. It is easy to see that any d-degenerate graph G can be partitioned into two $(d-1)$-degenerate graphs: construct a degenerate ordering and color the vertices in this order red or blue using the rule that a vertex v is colored red if it has less than d red neighbors; otherwise, color v blue. While this procedure leads to a partition into $(d-1)$-degenerate graphs, this partition need not be equitable. In fact, the only partition of the star S_m with m rays (which is 1-degenerate) into two independent sets (which are 0-degenerate) is that in which one set contains one vertex and the other contains the rest. Similarly, any partition of S_m into k 0-degenerate sets has one 1-element set and some set with at least m/k elements. In this section we show that if we have $d \geq 2$ and we allow for a third set, then we can provide equitability. This extends the Bollobás–Guy result [4] to arbitrary $d \geq 2$ and also provides a tool for obtaining equitable colorings that use few colors. Specifically, we will prove Theorem 3.

Theorem 7 (restatement of Theorem 3). *Let $k \geq 3$ and $d \geq 2$. Then every d-degenerate graph can be equitably partitioned into k $(d-1)$-degenerate graphs.*

Proof. We prove the result by contradiction, assuming that the above claim is false. Let G be a smallest (with respect to the number of vertices) counterexample to the theorem. Let $n = |V(G)|$. Then $n > dk$, because otherwise, any equitable vertex partition is good enough. A simple observation that forms the basis of the proof is the following.

Claim 4.1. *Let $v_1, v_2, \dots, v_m$ be a d-degenerate vertex ordering of a d-degenerate graph H. If $H - v_m$ has a k-partition $(W_1, \dots, W_k)$, where every W_i induces a $(d-1)$-degenerate subgraph, then among $W_1 + v_m, \dots, W_k + v_m$ at most one is not $(d-1)$-degenerate. Furthermore, if $W_i + v_m$ is not $(d-1)$-degenerate, then v_m has d neighbors and W_i contains all d neighbors of v_m.*

Proof. By the definition of a d-degenerate vertex ordering, the degree of v_m is at most d. If W_i has fewer than d neighbors of v_m, then we can append v_m to a $(d-1)$-degenerate ordering of W_i. □

Claim 4.2. *The minimum degree of G is d and n is divisible by k.*

Proof. Suppose that $n = k \cdot s + r$, where $1 \leq r \leq k$. We can choose a degenerate ordering of G such that the last vertex in the ordering, v_n, is a vertex of minimum degree. By the minimality of G, there exists an equitable k-partition $(W_1, \dots, W_k)$ of $V(G) - v_n$ into sets inducing $(d-1)$-degenerate graphs. Note that exactly $r-1$ of these sets have size $s+1$ and the remaining $k-r+1$ sets are of size s. Since $k-r+1 \geq 1$, there is at least one W_i of size s. If $\deg_G(v_n) \leq d-1$, then adding v_n to any set W_i of size s creates the desired equitable k-partition of G. This contradicts the choice of G and so we have that $\deg_G(v_n) \geq d$.

If k does not divide n, then we have $r < k$. This implies that there are $k-r+1 \geq 2$ sets of size s and, by Claim 4.1, we can add v_n to at least one of these sets of size s. Again, this contradicts the choice of G as a minimal counterexample and implies that k divides n. □

Given a vertex ordering $R = \{v_1, \dots, v_n\}$ of a graph H and an edge $e = v_i v_j \in E(H)$, we denote $l_R(e) = i$ and $r_R(e) = j$ if $i < j$. From all d-degenerate orderings of $V(G)$ choose a *special* ordering $U = (u_1, \dots, u_n)$, where the maximum index $l_U(e)$ of an edge $e \in E(G)$ is maximized. Let i_0 be the maximum of $l_U(e)$ over all the edges in the special ordering U. For convenience, we use U_i to denote the set $\{u_i, u_{i+1}, \dots, u_n\}$ for each i, $1 \leq i \leq n$.

Claim 4.3. *The vertex u_{i_0} is adjacent to u_i for every $i_0 < i \leq n$, and the set U_{i_0+1} is independent.*

Proof. The second part of the claim is directly implied by the definition of i_0. Suppose that for some $j > i_0$, the vertex u_j is not adjacent to u_{i_0}. Then all the neighbors of u_j are in $V(G) - U_{i_0}$. So moving u_j from its current position to just before

u_{i_0} creates another d-degenerate ordering of $V(G)$. In this ordering the maximum index of the left end of an edge is $i_0 + 1$, which contradicts the choice of the special ordering U. □

Now we are ready to prove the theorem.

Case 1. $i_0 \geq n-k+1$. Let $G' = G - U_{n-k+1}$. By the minimality of G, $V(G')$ has an equitable partition $(W_1, \dots, W_k)$ into sets inducing $(d-1)$-degenerate graphs. Now we attempt to consecutively add $u_{n-k+1}, u_{n-k+2}, \dots, u_n$ (in this order) so that (a) we add one vertex to every set, and (b) every new set still induces a $(d-1)$-degenerate graph. For vertices $u_{n-k+1}, u_{n-k+2}, \dots, u_{n-1}$ we can do this by Claim 4.1. Suppose that after adding vertices $u_{n-k+1}, u_{n-k+2}, \dots, u_{n-1}$, W_i is the only set to which no vertex has been added. The trick with u_n is that one of its neighbors is u_{i_0}, which has already been added to a set different from W_i. Thus u_n has at most $(d-1)$ neighbors in W_i and therefore the set $W_i \cup \{u_n\}$ still induces a $(d-1)$-degenerate graph.

Case 2. $i_0 \leq n-k$. Let $G'' = G - U_{i_0}$. By the minimality of G, $V(G'')$ has an equitable partition $(W_1, \dots, W_k)$ into sets inducing $(d-1)$-degenerate graphs. For $i > i_0$, call a set W_ℓ $1 \leq \ell \leq k$ *i-incompatible* if all $d-1$ neighbors of u_i different from u_{i_0} are in W_ℓ. By Claim 4.1, for every $i > i_0$, there could be at most one i-incompatible set. However, a set W_ℓ may be i-incompatible for several i. By Claim 4.1, u_{i_0} can be added to any one of at least $k-1$ sets among the W_i's. Let $S = \{W_i \mid 1 \leq i \leq k \text{ and } u_{i_0} \text{ can be added to } W_i\}$. There exists some set $W_{\ell'} \in S$ such that $W_{\ell'}$ is i-incompatible with at most $(n-i_0)/|S|$ values of $i > i_0$. Since $k \geq 3$, $|S| \geq 2$ and so $(n-i_0)/|S| \leq (n-i_0)/2$. Now add u_{i_0} to $W_{\ell'}$. Any u_i, $i > i_0$, for which $W_{\ell'}$ is i-incompatible, can be added to any set other than $W_{\ell'}$. Distribute such u_i's among sets other than $W_{\ell'}$ so that the sizes of new sets do not exceed $s = n/k$. The remaining u_i's can be added to any set. Thus, we add these in an arbitrary way so that the size of every W_l becomes $s = n/k$. □

Algorithm. The algorithm implied by the above proof is sketched here; the correctness of the algorithm follows from the proof. An equitable k-partition of a given n-vertex graph G is constructed recursively. If G contains a vertex of degree less than d or if n is not divisible by k, we construct a d-degenerate ordering of G and, assuming that v is the last vertex in this ordering, construct an equitable k-partition of $G - v$ and then add v to one of the k sets. Otherwise, we construct a special d-degenerate ordering U of G, referred to in the proof, as follows. Let L_0 be the set of vertices in G with degree at most d. If L_0 contains a pair of adjacent vertices, say u and v, then U is obtained by constructing an arbitrary d-degenerate ordering of $G - u - v$ and appending u and v to this. Otherwise, let L_1 be the set of vertices in $G - L_0$ with degree at most d. By definition, every vertex in L_1 has a neighbor in L_0. Find a vertex $v \in L_1$ with fewest neighbors in L_0. Let S denote the set of neighbors of v in L_0. U is obtained by constructing an arbitrary d-degenerate ordering of $G - v - S$ and appending v followed by vertices in S to this. Once U is constructed, we determine whether Case 1 (respectively, Case 2) of the proof applies and accordingly construct an equitable k-partition of $G' = G - U_{n-k+1}$ (respectively, $G'' = G - U_{i_0}$) and add vertices in U_{n-k+1} (respectively, U_{i_0}) to the sets in the partition. It is easy to see that $O(n^2)$ time suffices for the algorithm, though it seems likely that with more care this can be implemented in subquadratic time.

Remark. In [17], a list analogue of equitable coloring was considered. A *list assignment* L for a graph G assigns to each vertex $v \in V(G)$ a set $L(v)$ of allowable colors. An L-coloring of G is a proper vertex coloring such that for every $v \in V(G)$ the color on v belongs to $L(v)$. For example, when colors represent time periods and vertices are jobs, the list model incorporates the restriction that not all time periods are suitable for all jobs. A list assignment L for G is *k-uniform* if $|L(v)| = k$ for all $v \in V(G)$.

Given a k-uniform list assignment L for an n-vertex graph G, we say that G is *equitably L-colorable* if G has an L-coloring of G such that every color has at most $\lceil n/k \rceil$ vertices. A graph G is *equitably list k-colorable* if G is equitably L-colorable whenever L is a k-uniform list assignment for G.

Because some colors in the lists may occur rarely, one cannot ensure using each color, and most of the techniques previously used for ordinary equitable colorings do not work well for equitable list colorings. In particular, it is not absolutely clear how to adapt the proofs of Theorems 1 and 2 for equitable colorings. However, the idea of the proof of Theorem 3 could be adapted to prove its list version as follows.

THEOREM 8. *Let $k \geq 3$ and $d \geq 2$. Suppose that every vertex v of a d-degenerate graph G on n vertices is given a list $L(v)$ of k colors. Then the vertices of G can be colored from their lists in such a way that every color class induces a $(d-1)$-degenerate subgraph of G and contains at most $\lceil n/k \rceil$ vertices.*

REFERENCES

[1] B. Baker and E. Coffman, *Mutual exclusion scheduling*, Theoret. Comput. Sci., 162 (1996), pp. 225–243.

[2] J. Blazewicz, K. Ecker, E. Pesch, G. Schmidt, and J. Weglarz, *Scheduling Computer and Manufacturing Processes*, 2nd ed., Springer, Berlin, 2001.

[3] H. L. Bodlaender and F. V. Fomin, *Equitable Colorings on Graphs with Bounded Treewidth*, Tech. report UU-CS-2004-010, Universiteit Utrecht, Utrecht, The Netherlands.

[4] B. Bollobás and R. K. Guy, *Equitable and proportional colorings of trees*, J. Combin. Theory, Ser. B, 34 (1983), pp. 177–186.

[5] B.-L. Chen, K.-W. Lih, and P.-L. Wu, *Equitable coloring and the maximum degree*, European J. Combin., 15 (1994), pp. 443–447.

[6] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York, 1979.

[7] A. Hajnal and E. Szemerédi, *Proof of conjecture of Erdős*, in Combinatorial Theory and its Applications, Vol. II, P. Erdős, A. Rényi, and V. T. Sós, eds., North–Holland, Amsterdam, 1970, pp. 601–603.

[8] S. Irani and V. Leung, *Scheduling with conflicts, and applications to traffic signal control*, in Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, ACM, New York, 1996, pp. 85–94.

[9] S. Janson, *Large Deviations for Sums of Partly Dependent Random Variables*, preprint NI02024-CMP, Isaac Newton Institute, Cambridge, UK, 2002; Available online at http://www.newton.cam.ac.uk/preprints/NI02024.pdf.

[10] S. Janson, T. Luczak, and A. Ruciński, *Random Graphs*, Wiley-Interscience, New York, 2000.

[11] S. Janson and A. Ruciński, *The infamous upper tail*, Random Structures Algorithms, 20 (2002), pp. 317–342.

[12] F. Kitagawa and H. Ikeda, *An existential problem of a weight-controlled subset and its application to school timetable construction*, Discrete Math., 72 (1988), pp. 195–211.

[13] J. Komlós and M. Simonovits, *Szemerédi's regularity lemma and its applications in graph theory*, in Combinatorics: Paul Erdös Is Eighty, Vol. 2 (Keszthely, 1993), János Bolyai Math. Soc., Budapest, 1996, pp. 295–352.

[14] A. V. Kostochka, *Equitable colorings of outerplanar graphs*, Discrete Math., 258 (2002), pp. 373–377.

[15] A. V. Kostochka and K. Nakprasit, *Equitable colorings of k-degenerate graphs*, Combin. Probab. Comput., 12 (2003), pp. 53–60.

[16] S. V. Pemmaraju, K. Nakprasit, and A. V. Kostochka,, *Equitable colorings with constant number of colors*, in Proceedings of the 14th Annual SIAM-ACM Symposium on Discrete Algorithms, SIAM, Philadelphia, ACM, New York, pp. 458–459.

[17] A. V. Kostochka, M. J. Pelsmajer, and D. B. West, *A list analogue of equitable coloring*, J. Graph Theory, 44 (2003), pp. 166–177.

[18] J. Karrup and D. de Werra, *Chromatic optimisation: Limitations, objectives, uses, references*, European J. Oper. Res., 11 (1982), pp. 1–19.

[19] K.-W. Lih and P.-L. Wu, *On equitable coloring of bipartite graphs*, Discrete Math., 151 (1996), pp. 155–160.

[20] W. Meyer, *Equitable Coloring*, Amer. Math. Monthly, 80 (1973), pp. 143–149.

[21] S. V. Pemmaraju, *Equitable colorings extend Chernoff-Hoeffding bounds*, in Approximation, Randomization, and Combinatorial Optimization, Springer, Berlin, 2001, pp. 285–296.

[22] S. V. Pemmaraju, *Coloring outerplanar graphs equitably*, submitted. Available online at www.cs.uiowa.edu/~sriram/vita/vita.html.

[23] B. F. Smith, P. E. Bjorstad, and W. D. Gropp, *Domain Decomposition. Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, Cambridge, UK, 1996.

[24] A. Tucker, *Perfect graphs and an application to optimizing municipal services*, SIAM Rev., 15 (1973), pp. 585–590.

[25] H.-P. Yap and Y. Zhang, *The equitable Δ-colouring conjecture holds for outerplanar graphs*, Bull. Inst. Math. Acad. Sin., 25 (1997), pp. 143–149.

[26] H.-P. Yap and Y. Zhang, *Equitable colourings of planar graphs*, J. Combin. Math. Combin. Comput., 27 (1998), pp. 97–105.

SIAM J. Discrete Math.
Vol. 19, No. 1, pp. 96–108

COMPUTATION IN NOISY RADIO NETWORKS[*]

EYAL KUSHILEVITZ[†] AND YISHAY MANSOUR[‡]

Abstract. In this paper, we examine noisy radio (broadcast) networks in which every bit transmitted has a certain probability of being flipped. Each processor has some initial input bit, and the goal is to compute a function of these input bits. In this model, we show a protocol to compute any threshold function using only a linear number of transmissions.

Key words. radio networks, broadcast networks, noisy computation, threshold functions

AMS subject classifications. 68M15, 68W15

DOI. 10.1137/S0895480103434063

1. Introduction.

The influence of *noise* (or faults) on the complexity of computation has been studied in many contexts. In particular, researchers have been interested in *random* noise. In a typical scenario, it is assumed that the outcome of each operation is noisy, with some fixed probability p, and all the faults are independent. Usually, if t is the number of operations performed by the computation, then by repeating each operation $O(\log t)$ times and taking the majority of the results, one can ensure a constant probability of error at the cost of $O(t \log t)$ operations. It is desirable, however, to obtain a cost of $O(t)$ (i.e., to increase the utilized resources only by a constant factor compared to the nonnoisy case). In spite of the apparent simplicity of this noise model, it turns out that overcoming such a noise, with only a constant increase in the cost, may be nontrivial and in some cases impossible. Such a noise model has been studied in the contexts of circuits with noisy gates (e.g., [Neu56, DO77a, DO77b, Pip85, RS91]), decision trees with noisy nodes (e.g., [FRPU90, RS91, EP98]), communication complexity (e.g., [Sch96, RS94]), and others (e.g., [Tay68, Kuz73, Gác86, Spi96]).

In this paper, we consider noisy radio (broadcast) networks. The main feature of radio networks is that information is communicated by using broadcast; i.e., when a processor sends a message, all its neighbors receive it. There has been a considerable amount of attention given to this model. The most important problem, studied in this model, is the broadcast problem, where one processor wants to send some message to all the processors in the network (e.g., [CK85, CK87, CW87, ABLP91, BGI92, KM98, GM03] and references therein). In addition, general transformations have been developed to translate protocols from the radio model to the standard point-to-point network model [ABLP92]. All the above work assumes noise-free transmission. We consider the case of *random noise* as described above. To simplify things, we ignore the topology of the network and study the simplest network topology—a fully connected radio network; this allows us to focus on the issue of efficiently overcoming the noise.

[*] Received by the editors September 3, 2003; accepted for publication (in revised form) September 17, 2004; published electronically June 16, 2005. A preliminary version of this paper appeared in *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms,* San Francisco, CA, 1998, pp. 236–243.

http://www.siam.org/journals/sidma/19-1/43406.html

[†] Department of Computer Science, Technion, Haifa 32000, Israel (eyalk@cs.technion.ac.il, http://www.cs.technion.ac.il/∼eyalk).

[‡] Department of Computer Science, Tel-Aviv University, Ramat-Aviv, Israel (mansour@math.tau.ac.il).

This question of noisy radio (broadcast) networks was already studied by Gallager [Gal88]. More formally, Gallager considers the following setting: there are n processors in the network $P_1, \ldots, P_n$. Each processor P_i starts the protocol with an input bit b_i and each processor's goal is to compute the value of some function $f(b_1, \ldots, b_n)$. The basic operation in this model is a *broadcast* operation, in which one processor P_i sends a bit b and all processors hear it. We rule out encoding of information using silence by assuming that a processor has to send a bit whenever it is its turn to broadcast. Conceptually, one may view silence and nonsilence as encoding the values zero and one, and therefore they too should be subject to noise. (See a detailed discussion in section 2.2.)

Obviously, in the noise-free case, every function can be computed using n broadcast operations and n is also a lower bound for nondegenerate functions. In the *noisy* setting, when P_i broadcasts a bit, every processor P_j receives this bit with probability $1 - p$ and receives its complement, $\bar{b}$, with probability p. The question is what bounds can be proved in the noisy case. By the above arguments, $O(n \log n)$ broadcast operations suffice and $\Omega(n)$ broadcast operations are necessary. Gallager [Gal88] proved that in fact every function can be computed using $O(n \log \log n)$ broadcast operations.[1] It remained open whether one can achieve the desired $O(n)$ upper bound. In particular, Yao [Yao97] suggested concentrating on this open problem with respect to the family of *threshold functions* as a special case, which already seems to be difficult.[2]

In this paper, we answer this question in the affirmative for all *threshold functions* (this includes as special cases the functions AND, OR, and $MAJORITY$). Namely, we present a protocol to compute any such function using $O(n)$ broadcast operations. The correctness proof of our protocol uses some probabilistic analysis that might be useful in other settings as well. In particular, in our protocol the processors need not have an a priori knowledge of p, the noise probability (although for simplifying the presentation we start with the assumption that p is known; later, however, we show how to get rid of this assumption and use any upper bound $p' < 1/2$ on the true noise rate p). Also, as is usually required from protocols that run in noisy environments, the schedule in our protocol (i.e., the decision of which processor broadcasts next) is *oblivious*; i.e., it does not depend on messages received during the execution of the protocol (which are potentially noisy).

Subsequent work. Some relevant work was done after the conference version of our paper appeared. This subsequent work shows results of a nature similar to those presented in the conference version (i.e., computing simple functions on a radio network using a linear, or almost linear, number of broadcasts) in more complicated noise models. Feige and Kilian [FK00] consider an "adversarial" noise model. In this model, a processor receives a broadcasted bit correctly with some probability $1-p$. Then, the adversary may choose to correct the values of corrupted bits, but may not corrupt the values of correctly received bits. Very recently, Newman [New04] considered another noise model (which is also more general than the one considered in this paper), where the noise rate is not fixed. Instead, there is only a global upper bound p on the noise rate. Then, for each broadcast and for each receiver, with some probability $p^* \leq p$, the broadcast bit is received corrupted at the receiver (and, as usual, all errors are independent).

[1] Gallager's paper [Gal88] concentrates on the *parity* function; however, the protocol presented there can in fact deal with any function.

[2] A threshold function is defined using a parameter k. It returns 1 if $\sum_{i=1}^{n} b_i \geq k$ and 0 otherwise.

Organization. In section 2, we provide some necessary background from probability theory together with a precise definition of the model. In section 3, we present our protocol. We start with the slightly easier case, where the noise rate, p, is known to all processors. Later, in section 4, we extend our solution to handle the case in which p is unknown.

2. Preliminaries.

2.1. Probability theory. In this section, we provide some notation and facts from probability theory. Most of these are standard but certain facts are less commonly used in computer science.

Denote by $b(k; n, p)$ the probability of having *exactly* k successes in n independent trials, where each trial has probability p for success. That is, for integers $0 \le k \le n$ and a real number $0 \le p \le 1$,

$$b(k; n, p) \overset{\triangle}{=} \binom{n}{k} p^k (1-p)^{n-k}.$$

Denote by $B(k; n, p)$ the probability of having *at most* k successes in n trials, where each trial has probability p for success. That is,

$$B(k; n, p) \overset{\triangle}{=} \sum_{i=0}^{k} b(i; n, p) = \sum_{i=0}^{k} \binom{n}{i} p^i (1-p)^{n-i}.$$

For our proofs, we will need to estimate such binomial coefficients. Below are some useful (standard) facts that will help us in doing so (see, e.g., [SF96, p. 169]). We start with Stirling's formula

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \left(1 + O\left(\frac{1}{n}\right)\right).$$

Let k be either $\lfloor pn \rfloor$ or $\lceil pn \rceil$ (i.e., k is approximately the expected number of successes). Then,

$$b(k; n, p) = \frac{1}{\sqrt{2\pi p(1-p)n}} \left(1 + O\left(\frac{1}{n}\right)\right). \tag{2.1}$$

Finally, note that $b(k; n, p)$ is maximal for these values of k. That is, for any $k \le \lfloor pn \rfloor$,

$$b(k-1; n, p) \le b(k; n, p), \tag{2.2}$$

and similarly, for any $k \ge \lceil pn \rceil$,

$$b(k+1; n, p) \le b(k; n, p).$$

The following inequality, due to Chernoff [Che52], is a standard and useful fact. Let $X_1, \ldots, X_n$ be n independent 0-1 random variables with "success" probability p (i.e., $\Pr[X_i = 1] = p$). Then, the probability that $\sum_{i=1}^n X_i$ deviates significantly from its expected value (i.e., from pn) is small. More precisely, we have the following.

LEMMA 2.1. *Let $X_1, \ldots, X_n$ be n independent, identically distributed (i.i.d.) binary random variables, and let $p = \Pr[X_i = 1]i$ for all i. Then,*

$$\Pr\left[\sum_{i=1}^{n} X_i < (p - \lambda)n\right] \le e^{-\lambda^2 n/2p} \le e^{-\lambda^2 n/2}.$$

Suppose that we have n independent 0-1 random variables $X_1, \ldots, X_n$ with "success" probabilities $p_1, \ldots, p_n$ (respectively). Obviously, the expected value of $\sum_{i=1}^n X_i$ is $\mu \overset{\triangle}{=} \sum_{i=1}^n p_i$. Consider all possible choices for $p_1, \ldots, p_n$ such that their sum is μ. The following lemma, due to Hoeffding [Hof56], states that the probability of getting k successes (for k, which is smaller than the expectation) is maximal if all the p_i's are equal (i.e., each of them is μ/n). Formally, we have the following.

Lemma 2.2 (see [Hof56]). *Let X_i $(1 \le i \le n)$ be n independent, binary random variables. Let $\mu = E[\sum_{i=1}^n X_i]$. For any integer $k \le \mu - 1$,*

$$\Pr\left[\sum_{i=1}^{n} X_i \le k\right] \quad \le \quad B(k; n, \mu/n).$$

The above lemma allows us to transform situations in which we have several types of random variables with different probabilities into a more "uniform" setting.

The *median* of a probability distribution is a number x such that $\Pr(X \le x) \ge 1/2$ and $\Pr[X \ge x] \ge 1/2$. The following lemma [JS68] states that the median of a binomial distribution is very close to its expectation.

Lemma 2.3 (see [JS68]). *Let X_i be n i.i.d., binary random variables. Let $\mu = E[\sum_{i=1}^n X_i]$. The median of the distribution $X \overset{\triangle}{=} \sum_{i=1}^n X_i$ is either $\lfloor \mu \rfloor$ or $\lceil \mu \rceil$.*

We also use the following property of $b(k; n, p)$ that states that, when k is very close to the expectation (i.e., to pn), then the value of $b(k; n, p)$ is approximately $1/\sqrt{n}$ (for completeness, we prove this in the appendix).

Lemma 2.4. *Let $c \ge 0$ and $p \in (0, 1)$ be constants. Then, for sufficiently large n, we have*

$$b(\lfloor np \rfloor - c; n, p) = \Theta\left(\frac{1}{\sqrt{n}}\right).$$

2.2. Model. The model consists of n processors $P_1, \ldots, P_n$ that are communicating over a fully connected radio network (also called a *single-hop network* in the context of radio networks). When a certain processor P_i broadcasts a bit, all other processors (for simplicity, including itself) receive the bit corrupted by some noise. Formally, denote by $b_{i,k}$ the kth bit submitted by P_i during the execution of the protocol. When processor P_i broadcasts this bit $b_{i,k}$, then each processor P_j receives $b_{i,k} \oplus r_{i,j,k}$, where the $r_{i,j,k}$'s are i.i.d., binary random variables such that $\Pr[r_{i,j,k} = 1] = p < 1/2$.

Initially, each processor P_i has an input bit b_i. The goal of $P_1, \ldots, P_n$ is to evaluate some function f on their inputs (i.e., to compute the value $f(b_1, \ldots, b_n)$). We will be interested mainly in threshold functions, i.e., functions that depend only on whether the number of ones in the input exceeds a given threshold. We say that the protocol is *correct* if, with probability $\ge 1-\varepsilon$, *all* the processors output the correct value (where ε is any fixed constant, e.g., $\varepsilon = 0.01$).

We also require that the scheduling of the transmissions be *oblivious*; i.e., it can be fixed ahead of time independently of the input (and of previous trasmissions).[3] Whenever it is P_i's turn to broadcast according to the schedule, it *must* send a bit

[3] This is a common requirement for communication in noisy environments. The reason behind this requirement is that if the schedule depends on previous transmissions then, due to the noise, different processors get different information, which can result in chaotic situations.

and is not allowed to remain silent.[4] We emphasize that, since our paper is about proving upper bounds, putting extra requirements on the protocols only makes the results stronger.

The communication complexity measure is the number of broadcast operations performed (as a function of n). In particular, the obliviousness of the schedule implies that the number of transmissions is the same for all inputs and in all possible executions.

3. Computing the threshold function. In this section we present a protocol, `THRESHOLD(`k, ε`)`, which allows the n processors to decide whether the number of ones in the input, $\ell \stackrel{\triangle}{=} \sum_{i=1}^{n} b_i$, is at least as large as the "threshold parameter" k or not. If $\ell \geq k$, then the output should be 1 (i.e., "TRUE"); else ($\ell < k$), the output should be 0 (i.e., "FALSE"). We denote by ε the accuracy parameter. Recall that the correctness condition requires that, with probability at least $1 - \varepsilon$, *all* the processors have the correct value. We first assume that p, the noise rate, is known to all processors. Later, we show how to remove this assumption (section 4).

The basic idea is as follows: In the first round each processor P_i broadcasts its bit b_i for m_2 times (repeating each broadcast for $m_2 = O(1)$ times is intended to make sure that, for different values of ℓ, the expected number of ones received is significantly different); then P_i compares the number of ones that it received, A_i, to θ_k which is essentially the expected number of ones P_i should have received, given that the input contains exactly k ones. We denote by β_i the result of this comparison in processor P_i (i.e., $\beta_i = 1$ iff $A_i \geq \theta_k$). In the next two steps, each P_i computes γ_i, which is supposed to be the majority of the β_i's, and then computes δ_i, which is supposed to be the majority of the γ_i's and, as will be shown, is also supposed to be identical to the value of the function. One of the difficulties, of course, is that when the processors compute the majorities (i.e., the values γ_i, δ_i) the transmissions might, again, be corrupted by noise.

Intuitively, the main claim that we will prove is that the computed values $\beta_i, \gamma_i, \delta_i$ are expected to be more and more "biased" toward the correct output. More precisely, we start the protocol with a bias of the inputs to the correct output value that may be as small as $1/n$ (when we compare the border cases $\ell = k$ and $\ell = k - 1$). In round A, we set the parameters such that the bias of each processor increases to $O(1/\sqrt{n})$. In other words, we expect at least $n/2 + O(\sqrt{n})$ of the processors to have the correct output (i.e., to have β_i, that is, identical to the desired output value). In round B, we are taking the majority of n "noisy" votes β_i; now we can boost the number of processors that have the correct output (i.e., those processors for which γ_i is identical to the desired output value) to $n/2 + \Theta(n)$. Finally, in round C, since we already have a clear majority with the correct value, using a Chernoff bound, with overwhelming probability *all* the processors will have the correct output δ_i. Clearly, due to the noise, each step in this description might fail. In our analysis, we compute the failure probability of each round and sum them up in order to get the total probability of failure. This total probability of failure should be less than the accuracy parameter, ε.

[4] There are two widely studied models of broadcast/radio networks (see, e.g., [Tan81] for a discussion including a technological justification for each of the models): (a) a model that allows a processor to remain silent in its turn; and (b) a model that does not allow a processor to remain silent in its turn. Our work, as well as that of [Gal88] and many others, refers to the latter model. It is easy to see that in the former model all functions can be computed in $O(n)$ broadcasts even in the presence of noise (the reason is that in this model processors can encode information by silence and silence is assumed to be unaffected by noise).

The protocol THRESHOLD(k,ε) for processor P_i is as follows. (Recall that, for convenience, each time that a bit is broadcasted to all processor, it is also received by the sending processor.)

```
Initialize   Given $p$, $n$, and $k$,
             let $\theta_k = pnm_2 + k(1-2p)m_2 - m_1$,
             where $m_1 = \max\{2\ln(3/\varepsilon)/(c_2^2),\ \ 4(c_2(1-2p))^{-2}\}$
             and $m_2 = \lceil 2m_1/(1-2p) \rceil$
             (and $c_2$ is a constant to be fixed later).
Round A      Broadcast input $b_i$ (to all processors) $m_2$ times.
             Receive $m_2 n$ bits $\alpha_{i,j,m}$ (for $1 \le j \le n, 1 \le m \le m_2$).
             Let $A_i = \sum_{j,m} \alpha_{i,j,m}$.
             Let $\beta_i = 1$ iff $A_i \ge \theta_k$.
Round B      Broadcast (to all processors) $\beta_i$ once.
             Let $\gamma_i$ be the majority of the bits received. (Break ties arbitrarily.)
Round C      Broadcast (to all processors) $\gamma_i$ once.
OUTPUT       $\delta_i$—the majority of the bits received. (Break ties arbitrarily.)
```

3.1. Analysis. The following theorem is obvious from the code of the protocol.

THEOREM 3.1. *THRESHOLD(k,ε) is an oblivious protocol that requires only $O(n)$ broadcast operations.*

Our main task is to compute the error probability of the protocol. In the following series of lemmata, we establish the desired property at the end of each round; later, in Theorem 3.7, we combine these properties to establish that the error is bounded by ε.

LEMMA 3.2. *Let $\ell = \sum_{i=1}^{n} b_i$. Then,*

(1) *The random variables $A_1, \dots, A_n$ are i.i.d.;*

(2) *for all i, $E[A_i] = m_2 pn + (1-2p)m_2\ell$; and*

(3) *for all i, $|E[A_i] - \theta_k| \ge m_1 - 1$.*

Proof. The random variables A_i are identically distributed since, for every processor P_i, the value of A_i is defined in the same way (in particular, the noise rate is the same for all messages and, for convenience, every processor sends its input bit to itself as well). The independence of the A_i's is by the independence of the noise in our model.

As for the expectation, each of the ℓ bits b_j such that $b_j = 1$ contributes to the expectation $1-p$ in each of the m_2 times it is transmitted (since $1-p$ is the probability that a "1" bit is received as "1"), and each of the $n-\ell$ bits b_j such that $b_j = 0$ contributes to the expectation p in each of the m_2 times it is transmitted (since p is the probability that a "0" bit is received as "1"). Altogether

$$\begin{aligned} E[A_i] &= m_2(\ell(1-p) + (n-\ell)p) \\ &= m_2(\ell(1-2p) + np) \\ &= m_2 np + (1-2p)m_2\ell. \end{aligned}$$

The third inequality follows from the fact that for $\ell = k$ (using the definition of θ_k), we have $E[A_i] - \theta_k = m_1$ and that, for $\ell = k-1$, we have (using the definition of m_2)

$$|E[A_i] - \theta_k| = |m_1 - (1-2p)m_2| \ge |m_1 - (2m_1 + 1)| = m_1 - 1. \qquad \square$$

Note that we chose the value θ_k in the protocol in a way that it is at least $m_1 - 1$ away from the expected value of A_i. The next lemma claims that by comparing A_i

and θ_k we can know, with probability slightly better than 1/2, whether ℓ (the number of 1's in the input) is at least k (the threshold). More precisely, for any given i, the value β_i, computed in round A, is correct (i.e., it is equal to the value of the function, which is 1 iff $\ell \geq k$) with a probability that is larger than 1/2 by at least $\Omega(\sqrt{m_1/n})$.

LEMMA 3.3. *Let $\ell = \sum_{i=1}^{n} b_i$ and $p < 1/2$ be a constant. For some constant c_1 and for all i,*

(1) *if $\ell \geq k$, then $\Pr[A_i > \theta_k] > 1/2 + c_1\sqrt{m_1/n}$; and*

(2) *if $\ell < k$, then $\Pr[A_i < \theta_k] > 1/2 + c_1\sqrt{m_1/n}$.*

Proof. By Lemma 3.2, the expected value of A_i is $\mu_\ell = pnm_2 + (1-2p)m_2\ell$. For $k \leq \ell$, we have $\theta_k < \mu_k \leq \mu_\ell$, and, moreover, $\mu_\ell - \theta_k \geq m_1 - 1$. For simplicity assume that θ_k is not integral, i.e., $\lfloor\theta_k\rfloor \neq \lceil\theta_k\rceil$. By Lemma 2.2,

$$\Pr[A_i > \theta_k] \quad \geq \quad 1 - B(\lfloor\theta_k\rfloor; N, r_\ell), \tag{3.1}$$

where $N = m_2 n$ and $r_\ell = \mu_\ell/N$. (Note that $r_\ell = (1-\ell/n)p + \ell/n(1-p)$ is always between p and $1-p$.) Unfortunately, it seems that using a standard Chernoff bound (e.g., Lemma 2.1) would not give us much (since trying to distinguish "close" probabilities, as is the case here, would fail if we use a small sample size). On the other hand, it is clear that $\Pr[A_i > \theta_k]$ is minimized when the difference between k and ℓ is minimized, therefore we can assume, without loss of generality, that $\ell = k$. In addition, we need a refined analysis that bounds the individual binomial coefficients. By Lemma 2.3, the median for the binomial distribution is either $\lfloor\mu_\ell\rfloor$ or $\lceil\mu_\ell\rceil$. We can bound the desired probability,

$$B(\lfloor\theta_k\rfloor; N, r_k) = B(\lceil\mu_k\rceil - 1; N, r_k) - (B(\lceil\mu_k\rceil - 1; N, r_k) - B(\lfloor\theta_k\rfloor; N, r_k)) \tag{3.2}$$

$$\leq 1/2 - \sum_{j=1}^{\lceil\mu_k\rceil - \lfloor\theta_k\rfloor} b(\lceil\mu_k\rceil - j; N, r_k), \tag{3.3}$$

where we use here the fact that $B(\lceil\mu_k\rceil - 1; N, r_k) \leq 1/2$ (by the definition of median). Note that, by the definition of θ_k, we have $\lceil\mu_k\rceil - \lfloor\theta_k\rfloor \geq m_1$. By applying Lemma 2.4, for each $1 \leq j \leq \lceil\mu_k\rceil - \lfloor\theta_k\rfloor$, we get

$$b(\lceil\mu_k\rceil - j; N, r_k) = \Theta\left(\frac{1}{\sqrt{N}}\right). \tag{3.4}$$

Therefore, by substituting (3.3) and (3.4) into (3.1), and by the choice of parameters ($N = m_2 n$ and $m_2 = \Theta(m_1)$), we get

$$\Pr[A_i > \theta_k] \geq 1 - \left(\frac{1}{2} - m_1 \cdot \Theta\left(\frac{1}{\sqrt{m_2 n}}\right)\right)$$
$$= \frac{1}{2} + c_1\sqrt{m_1/n},$$

which completes the proof for the case when $k \leq \ell$.

The proof for the case when $k > \ell$ is similar. □

LEMMA 3.4. *At the end of round A, with probability at least $1 - e^{-c_2^2 m_1/2}$, at least $n/2 + c_2\sqrt{m_1 n}$ of the values β_i are correct, where $c_2 = c_1/2$.*

Proof. By Lemma 3.3, for each processor P_i, the probability that β_i is correct is at least $1/2 + c_1\sqrt{m_1/n}$. Consider the probability that the number of correct β_i's is smaller than $n/2 + c_2\sqrt{nm_1}$. By Lemma 2.1, this probability is at most $e^{-((c_1-c_2)\sqrt{m_1/n})^2 n/2} = e^{-c_2^2 m_1/2}$. □

LEMMA 3.5. *Assume that, at the beginning of round B, there are at least $n/2 + c_2\sqrt{nm_1}$ of the β_i's with the correct value. Then, at the end of round B, with probability at least $1 - e^{-q^2 n/2}$, at least $(1 - 2q)n$ of the processors have the correct value, where $q = e^{-(c_2(1-2p))^2 m_1/2} < e^{-2}$, and $m_1 > 4(c_2(1 - 2p))^{-2}$.*

Proof. Each γ_i is a random variable, which is the majority of n random variables, β_i, corrupted by noise rate p. By the assumption of the lemma, at least $n/2+c_2\sqrt{nm_1}$ of the β_i's are correct and at most $n/2-c_2\sqrt{nm_1}$ are incorrect. We want to bound the probability that a processor, after the noise is applied to the broadcast messages, gets more incorrect values than correct. Using Lemma 2.2, we can bound the probability of this event by $B(n/2; n, r)$, where

$$\begin{aligned} r &= \left(\frac{1}{2} + c_2\sqrt{\frac{m_1}{n}}\right) \cdot (1 - p) + \left(\frac{1}{2} - c_2\sqrt{\frac{m_1}{n}}\right) \cdot p \\ &= \frac{1}{2} + c_2(1 - 2p)\sqrt{\frac{m_1}{n}}. \end{aligned}$$

Using Lemma 2.1, we can now bound this probability by $e^{-(c_2(1-2p)\sqrt{m_1/n})^2 n/2} = e^{-(c_2(1-2p))^2 m_1/2} = q$. Therefore, each γ_i has a probability at least $1 - q$ of being correct. By the choice of m_1, we ensure that $q < e^{-2}$.

We need to bound the probability that the number of correct γ_i's is significantly smaller than $(1-q)n$ (which is a lower bound on the expected number of correct γ_i's). Using again Lemma 2.1, with probability at least $1 - e^{-q^2 n/2}$, at least $(1 - 2q)n$ of the γ_i's have the correct value. □

LEMMA 3.6. *Given that at the beginning of round C at least $(1 - 2q)n$ (for $q < e^{-2}$) of the bits γ_i have the correct value, then at the end of round C, with probability $1 - ne^{-((1/2-2q)(1-2p))^2 n/2}$, all the processors output the correct value.*

Proof. Again, using Lemma 2.2, we can bound the probability that a given output δ_i is incorrect by $B(n/2; n, r)$, where r is the average probability of success. That is,

$$r = (1 - 2q)(1 - p) + 2qp = \frac{1}{2} + \left(\frac{1}{2} - 2q\right)(1 - 2p).$$

Using Lemma 2.1, we bound this probability by at most $e^{-((1/2-2q)(1-2p))^2 n/2}$ for each processor. Summing over all the processors, we get the desired bound. □

We can now establish the correctness of the protocol with respect to the accuracy parameter ε.

THEOREM 3.7. *Let $\ell = \sum_{i=1}^{n} b_i$. For any k and ε and sufficiently large n, when running the protocol* THRESHOLD(k, ε) *with probability $1 - \varepsilon$, if $\ell \geq k$, then each processor P_i has $\delta_i = 1$, and if $\ell < k$, then each processor P_i has $\delta_i = 0$.*

Proof. By Lemma 3.4, after round A, with probability at least $1 - \varepsilon/3$, at least $n/2 + c_2\sqrt{m_1 n}$ of the values β_i are correct (since $m_1 \geq (2\ln(3/\varepsilon))/(c_2^2)$).

By Lemma 3.5, if the protocol does not fail in round A then, with probability at least $1 - e^{-ne^{-4}/2}$, we have at least $(1 - 2e^{-2})n$ processors with the correct value. Note that the error probability in this round goes to zero as n increases and therefore, for sufficiently large n, the failure probability is less than $\varepsilon/3$.

By Lemma 3.6, if the protocol does not fail in rounds A and B then, with probability at least $1-ne^{-(1/2-2e^{-2})^2 n/2}$, all the processors output the correct value. Again, if we choose n sufficiently large, then this will be less than $\varepsilon/3$. Hence, the total failure probability is bounded by ε.

The parameter m_1 has two constraints, which are independent of n. The first appears in Lemma 3.5 and requires that $m_1 > 4(c_2(1-2p))^{-2}$. The second constraint, from Lemma 3.4, requires that $m_1 \geq (2\ln(3/\varepsilon))/(c_2^2)$. Our choice of m_1 meets both bounds. $\square$

4. Unknown noise rate (p). In this section, we present a protocol for computing threshold functions even in the case where the noise rate is unknown. A simple approach for transforming the algorithm of the previous section into the case where the noise rate p is unknown is the following. First, estimate the noise rate p and then use this estimate, $\hat{p}$, in place of the real noise rate p. When considering the algorithm, one can observe that an error of $O(1/n)$ in $\hat{p}$ would increase θ_k by only a constant and hence will not matter. Unfortunately, the difference between p and $\hat{p}$ would be much larger (in all estimation methods that we know of that use only $O(n)$ broadcasts); this would cause this approach to fail. (For example, assume that initially each processor sends a zero bit, and then the number of ones received by each processor serves as an estimate of the error rate. For a single processor, the expected error would be $\Theta(1/\sqrt{n})$, which is much too large. If we could combine the error estimates computed at all the processors, we would get a much better estimate because in $\Theta(n)$ broadcasts there are $\Theta(n^2)$ received bits. But we do not know how to compute this with $O(n)$ broadcasts.) An additional difficulty in applying this approach is that we may lose the obliviousness of the protocol: we need all the processors to have the *same* value for $\hat{p}$, as it influences the value of m_2 and hence the schedule of the protocol.

Here we take a different approach to overcoming the unknown noise rate. In fact, we never attempt to estimate the noise rate p; instead, we only assume that the processors are given some upper bound p' on the noise rate, i.e., a value p' such that $p < p' < 1/2$. (Clearly, having such a bound is a more realistic assumption than actually knowing the exact noise rate p; such a bound may sometimes be achieved by empirical methods or just as an assumption on the environment in which the protocol runs.) The idea is to replace the constant θ_k with a random variable Θ_k, such that $E[\Theta_k] = \theta_k$, and to reprove Lemma 3.3 with Θ_k instead of θ_k. (We need to get a nontrivial bound on the noise, p', since a direct estimate of the noise rate would incur an error of $\Omega(1/\sqrt{n})$, as explained above. Also, our running time would depend on the distance of the bound p' from 1/2.)

First, we define the following round of broadcasts. Each of the k processors $P_1, \ldots, P_k$ sends a bit 1, and each of the $n-k$ processors $P_{k+1}, \ldots, P_n$ sends a bit 0. Each processor repeats this m_2 times, where m_2 is defined using p' instead of p, i.e., $m_2 = \lceil 2m_1/(1-2p') \rceil$ and $m_1 = (8 + \ln 1/\varepsilon)/(c_2(1-2p'))^2$. Processor P_i receives $m_2 n$ bits $\eta_{i,j}$ and sets $\Theta_{k,i} = \sum_{j=1}^{m_2 n} \eta_{i,j} - m_1$ (each processor P_i has its own value $\Theta_{k,i}$; note that this value is only used by P_i in the computation of β_i and has no influence on the schedule).

We observe that the values $\Theta_{k,i}$ (for $1 \leq i \leq n$) are i.i.d. In addition, the expectation of $\Theta_{k,i}$ is

$$\begin{aligned} E[\Theta_{k,i}] &= [(1-p)m_2 k + p m_2 (n-k)] - m_1 \\ &= m_2 p n + m_2 k (1-2p) - m_1 \\ &= \theta_k. \end{aligned}$$

It remains to show a lemma analogous to Lemma 3.3.

LEMMA 4.1. *Let $\ell = \sum_i b_i$ and $p < 1/2$ be a constant. For some constant c_3,*
(1) *if $\ell \geq k$, then $\Pr[A_i > \Theta_{k,i}] > 1/2 + c_3\sqrt{m_1/n}$; and*
(2) *if $\ell < k$, then $\Pr[A_i < \Theta_{k,i}] > 1/2 + c_3\sqrt{m_1/n}$.*

Proof. First, consider the case $\ell \geq k$. Clearly, the probability that A_i is larger than $\Theta_{k,i}$ is minimized when $\ell = k$. In this case, the random variables A_i and $\Theta_{k,i}+m_1$ are identical. To see this, we introduce a *permutation* σ of the processors as follows: we match each processor P_i, for $1 \leq i \leq k$, in the computation of $\Theta_{k,i}$, to $P_{\sigma(i)}$ which is one of the k processors having input 1 in the computation of A_i. Similarly, we match each processor P_i, for $k+1 \leq i \leq n$, in the computation of $\Theta_{k,i}$, to $P_{\sigma(i)}$, which is one of the remaining $n-k$ processors (those that have input 0) in the computation of A_i. Therefore, the random variables A_i and $\Theta_{k,i}+m_1$ are identical, and we conclude that

$$\Pr[A_i \geq \Theta_{k,i} + m_1] \geq \frac{1}{2}.$$

For the proof we will use the identity

$$\Pr[A_i > \Theta_{k,i}] = \Pr[A_i \geq \Theta_{k,i} + m_1] + \Pr[\Theta_{k,i} + m_1 > A_i > \Theta_{k,i}].$$

The first term is at least half, as explained above, so now we are interested in bounding the second term, which will give the bias that A_i has over $\Theta_{k,i}$.

Consider pairs of messages seen by processor P_i: in each pair, one message is from the computation of $\Theta_{k,i}$ and the other is the corresponding message from the computation of A_i (according to σ). That is, for some m ($1 \leq m \leq m_2$), each pair is of the form $\alpha_{i,j,m}$ and $\eta_{i,\sigma(j),m}$. For each pair, the messages $\alpha_{i,j,m}$ and $\eta_{i,\sigma(j),m}$ are i.i.d. Therefore, given that $\alpha_{i,j,m} \neq \eta_{i,\sigma(j),m}$, the probability that $1 = \alpha_{i,j,m} > \eta_{i,\sigma(j),m} = 0$ is exactly $1/2$ (by symmetry). Also, the probability that $\alpha_{i,j,m} \neq \eta_{i,\sigma(j),m}$ is $2p(1-p)$. Let t be the number of pairs in which $\alpha_{i,j,m} \neq \eta_{i,\sigma(j),m}$. Then, it follows that the expected value of t is $2p(1-p)m_2 n$. Using Lemma 2.1, with probability $1-2e^{-(p(1-p))^2 m_2 n/2} = 1-e^{-c'n}$, we have $t \in [p(1-p)m_2 n, 3p(1-p)m_2 n]$.

Let $\Delta \overset{\triangle}{=} |\sum_{i,j,m} \alpha_{i,j,m} - \sum_{i,j,m} \eta_{i,j,m}|$. Given that there are exactly t pairs, where elements of each pair are different, the probability that $\Delta \leq m_1$ is $O(m_1/\sqrt{t})$. Assuming that t is in the range $[p(1-p)m_2 n, 3p(1-p)m_2 n]$, we have that the probability is at least

$$c_4\sqrt{\frac{(1-2p')m_1}{p(1-p)n}} = c_5\sqrt{\frac{m_1}{n}}$$

for a constant p. Therefore,

$$\begin{aligned}\Pr[A_i > \Theta_{k,i}] &= \Pr[A_i \geq \Theta_{k,i} + m_1] + \Pr[\Theta_{k,i} + m_1 > A_i > \Theta_{k,i}] \\ &> \frac{1}{2} + c_5\sqrt{\frac{m_1}{n}} - e^{-c'n} \\ &= \frac{1}{2} + c_3\sqrt{\frac{m_1}{n}}.\end{aligned}$$

This concludes the case that $\ell \geq k$.

Now, assume that $\ell < k$. Again, the probability would be minimized when $\ell = k-1$. In this case we use σ to match only $n-1$ processors with identical inputs, and we are left with a pair of processors: one sends 1 (in the computation of $\Theta_{k,i}$) and the other sends 0 (in the computation of A_i). Without loss of generality, and to simplify the notation, we assume that processor P_1, which broadcasts 1 in the computation of $\Theta_{k,i}$, has input 0 in the protocol and is not matched by σ. Let the absolute difference between the sums of the η's and the α's in the $n-1$ matched processors be Δ, i.e., $\Delta \overset{\triangle}{=} |(\sum \eta_{i,j,m}) - (\sum \alpha_{i,j,m})|$ (where each sum is over $1 \le i \le n, 2 \le j \le n, 1 \le m \le m_2$). We will bound our event using the identity

$$\begin{aligned}\Pr[A_i < \Theta_{k,i}] = &\Pr[A_i < \Theta_{k,i} | \Delta \le m_1] \cdot \Pr[\Delta \le m_1] \\ + \;\; &\Pr[A_i < \Theta_{k,i} | \Delta > m_1] \cdot \Pr[\Delta > m_1],\end{aligned}$$

where $m_1 = (1-2p')m_2/2$. Intuitively, when Δ is less than m_1, the pair of unmatched processors (which, by the above assumption, is processor P_1 in both cases) "decides" the outcome.

Let t be the number of pairs in which $\alpha_{i,j,m}$ and $\eta_{i,\sigma(j),m}$ differ. Then, the probability that $\Delta \le m_1$ is $c_6 m_1/\sqrt{t}$ for some constant c_6. Again, with probability $1 - e^{-c'n}$, we have that $t \in [m_2 np(1-p), 4m_2 np(1-p)]$. Therefore,

$$\Pr[\Delta \le m_1] \ge c_6 \frac{m_1}{\sqrt{t}} - e^{-c'n} = c_7 \sqrt{\frac{m_1}{n}} = q$$

for a constant p.

Now consider the pair of unmatched processors. The expected difference in the unmatched processors is at least $(1-2p')m_2 \ge 2m_1$. Therefore, the difference in the unmatched processors is more than m_1, with probability at least $1 - 2e^{-(1-2p')^2 m_2/2} > 2/3$, for $m_1 > (\ln 6)/(1-2p')$. This implies that, given that $\Delta \le m_1$, the probability that $\Theta_{k,i} > A_i$ is strictly bounded away from half. Namely,

$$\Pr[\Theta_{k,i} > A_i | \Delta \le m_1] \ge \frac{2}{3}.$$

Given that $\Delta > m_1$, it is equally likely to favor A_i or $\Theta_{k,i}$. Since the bias in the unmatched pair is in favor of $\Theta_{k,i}$ we have

$$\Pr[\Theta_{k,i} > A_i | \Delta \le m_1] \ge 1/2.$$

To conclude,

$$\begin{aligned}&\Pr[A_i < \Theta_{k,i}] \\ &\quad = \Pr[A_i < \Theta_{k,i} | \Delta \le m_1] \cdot \Pr[\Delta \le m_1] + \Pr[A_i < \Theta_{k,i} | \Delta > m_1] \cdot \Pr[\Delta > m_1] \\ &\quad \ge \frac{2}{3}\, q + \frac{1}{2}(1-q) \\ &\quad = \frac{1}{2} + \frac{q}{6}.\end{aligned}$$

This concludes the case $k > \ell$. □

Once we have established Lemma 4.1, the proof is the same as before. Therefore, we have established the following theorem.

THEOREM 4.2. *There exists a protocol with $O(n)$ broadcasts that computes any threshold function, even if the actual noise rate p is unknown and only an upper bound p' is available.*

Appendix. Proof of Lemma 2.4. By definition,

$$b(k;n,p) = \frac{k+1}{n-k} \cdot \frac{1-p}{p} \cdot b(k+1;n,p).$$

For $k = \lfloor pn \rfloor - j$, we get

$$\begin{aligned} b(\lfloor pn \rfloor - j; n, p) &= \frac{\lfloor pn \rfloor - j + 1}{n - \lfloor pn \rfloor + j} \cdot \frac{1-p}{p} \cdot b(\lfloor pn \rfloor - j + 1; n, p) \\ &\leq \frac{pn - j + 1 - p\lfloor pn \rfloor + pj - p}{np - p\lfloor pn \rfloor + pj} \cdot b(\lfloor pn \rfloor - j + 1; n, p) \\ &= \left(1 - \frac{j + p - 1}{np - p\lfloor pn \rfloor + pj}\right) \cdot b(\lfloor pn \rfloor - j + 1; n, p). \end{aligned}$$

Also, using the Stirling formula (see (2.1)),

$$b(\lfloor pn \rfloor; n, p) = \frac{1 + O(1/n)}{\sqrt{2\pi p(1-p)n}}.$$

Therefore, by applying the above (and (2.2)) we get

$$\begin{aligned} b(\lfloor pn \rfloor - c; n, p) &= \left(\prod_{j=1}^{c} \left(1 - \frac{j+p-1}{np - p\lfloor pn \rfloor + pj}\right)\right) \cdot b(\lfloor pn \rfloor; n, p) \\ &= \left(1 - \frac{\Theta(c^2)}{2p(1-p)n} + O\left(\frac{1}{n^2}\right)\right) \frac{1 + O(1/n)}{\sqrt{2\pi p(1-p)n}} \\ &= \Theta\left(\frac{1}{\sqrt{n}}\right) \end{aligned}$$

for constants p and c.

REFERENCES

[ABLP91] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg, *A lower bound for radio broadcast*, J. Comput. System Sci., 43 (1991), pp. 290–298.

[ABLP92] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg, *Single round simulation on radio networks*, J. Algorithms, 13 (1992), pp. 188–210.

[BGI92] R. Bar-Yehuda, O. Goldreich, and A. Itai, *On the time-complexity of broadcast in multi-hop radio networks: An exponential gap between determinism and randomization*, J. Comput. System Sci., 45 (1992), pp. 104–126.

[Che52] H. Chernoff, *A measure of the asymptotic efficiency for tests of a hypothesis based on the sum of observations*, Ann. Math. Statist., 23 (1952), pp. 493–509.

[CK85] I. Chlamtac and S. Kutten, *On broadcasting in radio networks—problem analysis and protocol design*, IEEE Trans. Comm., 33 (1985), pp. 1240–1246.

[CK87] I. Chlamtac and S. Kutten, *Tree-based broadcasting in multihop radio networks*, IEEE Trans. Comm., 36 (1987), pp. 1209–1223.

[CW87] I. Chlamtac and O. Weinstein, *The wave expansion approach to broadcasting in multihop radio networks*, in Proceedings of the Sixth Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE, Piscataway, NJ, 1987, pp. 874–881.

[DO77a] R. L. Dobrushin and S. I. Ortyukov, *Lower bound for the redundancy of self-correcting arrangements of unreliable functions*, Probl. Inform. Transm., 13 (1977), pp. 59–65.

[DO77b] R. L. Dobrushin and S. I. Ortyukov, *Upper bound for the redundancy of self-correcting arrangements of unreliable functions*, Probl. Inform. Transm., 13 (1977), pp. 203–218.

[EP98] W. Evans and N. Pippenger, *Average-case lower bounds for noisy Boolean decision trees*, SIAM J. Comput., 28 (1998), pp. 433–446.

[FK00] U. Feige and J. Kilian, *Finding OR in a noisy broadcast network*, Inform. Process. Lett., 73 (2000), pp. 69–75.

[FRPU90] U. Feige, P. Raghavan, D. Peleg, and E. Upfal, *Computing with noisy information*, SIAM J. Comput., 23 (1994), pp. 1001–1018.

[GM03] I. Gaber and Y. Mansour, *Broadcast in radio networks*, J. Algorithms, 46 (2003), pp. 1–20.

[Gác86] P. Gács, *Reliable computation with cellular automata*, J. Comput. System Sci., 32 (1986), pp. 15–78.

[GG94] P. Gács and A. Gál, *Lower bounds for the complexity of reliable Boolean circuits with noisy gates*, IEEE Trans. Inform. Theory, 40 (1994), pp. 579–583.

[Gal88] R. G. Gallager, *Finding parity in a simple broadcast network*, IEEE Trans. Inform. Theory, 34 (1988), pp. 176–180.

[Hof56] W. Hoeffding, *On the distribution of the number of successes in independent trials*, Ann. Math. Statist., 27 (1956), pp. 713–721.

[JS68] K. Jogdeo and S. M. Samuels, *Monotone convergence of binomial probabilities and a generalization of Ramanujan's equation*, Ann. Math. Statist., 39 (1968), pp. 1191–1195.

[KM98] E. Kushilevitz and Y. Mansour, *An $\Omega(D\log(N/D))$ lower bound for broadcast in radio networks*, SIAM J. Comput., 27 (1998), pp. 702–712.

[Kuz73] A. V. Kuznetsov, *Information storage in a memory assembled from unreliable components*, Probl. Inform. Trans., 9 (1973), pp. 254–264.

[Neu56] J. von Neumann, *Probabilistic logics and the synthesis of reliable organisms from unreliable components*, in Automata Studies, C. E. Shannon and J. McCarthy, eds., Princeton University Press, Princeton, NJ, 1956, pp. 43–98.

[New04] I. Newman, *Computing in fault tolerance broadcast networks*, in Proceedings of the 19th Annual IEEE Conference on Computational Complexity (CCC), 2004, pp. 113–122.

[Pip85] N. Pippenger, *On networks of noisy gates*, in Proceedings of the 26th Annual Symposium on Foundations of Computer Science, IEEE, Los Alamitos, CA, 1985, pp. 30–36.

[RS91] R. Reischuk and B. Schmeltz, *Reliable computation with noisy circuits and decision trees–A general $n\log n$ lower bound*, in Proceedings of the 32nd Annual Symposium on Foundations of Computer Science, IEEE, Los Alamitos, CA, 1991, pp. 602–611.

[RS94] S. Rajagopalan and L. J. Schulman, *A Coding theorem for distributed computing*, in Proceedings of the 26th Annual Symposium on the Theory of Computing, ACM, New York, 1994, pp. 790–799.

[Sch96] L. J. Schulman, *Coding for interactive communication*, Special issue on Codes and Complexity, IEEE Trans. Inform. Theory, 42(6), part I, (1996), pp. 1745–1756.

[SF96] R. Sedgewick and P. Flajolet, *An Introduction to the Analysis of Algorithms*, Addison-Wesley, New York, 1996.

[Spi96] D. A. Spielman, *Highly fault-tolerant parallel computation*, in Proceedings of the 37th Annual Symposium on Foundations of Computer Science, IEEE, Los Alamitos, CA, 1996, pp. 154–163.

[Tan81] A. S. Tanenbaum, *Computer Networks*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[Tay68] M. G. Taylor, *Reliable information storage in memories designed from unreliable components*, Bell System Tech. J., 47 (1968), pp. 2299–2337.

[Yao97] A. C. Yao, *On the complexity of communication under noise*, invited talk at the 5th Israel Symposium on Theory of Computing and Systems, 1997.

SIAM J. Discrete Math.
Vol. 19, No. 1, pp. 109–121

THE COFFMAN–GRAHAM ALGORITHM OPTIMALLY SOLVES UET TASK SYSTEMS WITH OVERINTERVAL ORDERS*

MARC CHARDON† and AZIZ MOUKRIM†

Abstract. Scheduling of unit execution time (UET) task systems on parallel machines with minimal schedule length is known to be NP-complete. The problem is polynomially solvable for some special cases. For a fixed number of parallel machines $m > 2$, the complexity of the problem is still open, but the problem becomes NP-hard if m is arbitrary. In this paper we characterize a new order class that properly contains quasi-interval orders and we prove that the Coffman–Graham algorithm yields optimal schedules for this new class on any number of machines. Finally, some extensions are discussed for a larger order class and for scheduling in the presence of unit communication delays.

Key words. nonpreemptive profile scheduling, list scheduling, makespan, quasi-interval order

AMS subject classifications. 90B35, 06A06

DOI. 10.1137/S0895480101394999

1. Introduction.

We consider a set X of n partially ordered unitary *tasks* represented by a partial order $G = (X, \prec)$, also called a *precedence graph*. Tasks must be performed by a number of identical processors, which may vary in time without violating the precedence constraints. It is required that if $x \prec y$, then the execution of task y cannot begin until the execution of x has been completed. At each time slot the number of executed tasks cannot exceed the number of available processors, referred to as a *profile*. In nonpreemptive scheduling problems, it is assumed that a task, once started, is executed to completion. Our goal is to determine a schedule that minimizes the time taken to finish all the tasks. Such a schedule is said to be *optimal*. The maximum number of processors available at any time is called the *breadth* and is denoted m. A profile is said to be *straight* if the number of available processors is the same at any time.

Ullman shows in [17] that for general m this problem is NP-complete.

Polynomial algorithms have been developed for optimally solving some special cases. In the case of a straight profile, Hu [10] gives an $O(n)$ algorithm for scheduling inforests and outforests. Also, for a straight profile and a fixed breadth m, Dolev and Warmuth [6] give an $O(n^{2m-2} \log n)$ algorithm for opposing forests. Scheduling of interval orders can be done in linear time [15, 12]. A generalization of this result to quasi-interval orders is given in [13]. If the number of machines is limited to 2, this problem is solvable for arbitrary precedence constraints [3, 12]. Also, there exists an $O(n^{2l})$ algorithm for precedence graphs of bounded width l [1, 16]. For an arbitrary profile with a fixed breadth m, Dolev and Warmuth [6] provide an $O(n^{m-1})$ algorithm for scheduling level orders, an $O(n^{m-1})$ algorithm for inforests, and an $O(n^{m-1} \log n)$ algorithm for outforests. Also, Dolev and Warmuth prove in [5] that if the breadth of the profile is fixed and the height of the precedence graph is bounded by a constant

*Received by the editors September 14, 2001; accepted for publication (in revised form) August 18, 2004; published electronically June 16, 2005. This work was supported by "Conseil Régional de Picardie" and FSE. A preliminary version of this work appears in *Proceedings of the* 15*th International Parallel and Distributed Processing Symposium*, (April 23–27, 2001, San Francisco, CA), IEEE, Piscataway, NJ, 2001, pp. 1385–1392.

http://www.siam.org/journals/sidma/19-1/39499.html

†UMR CNRS 6599 HeuDiaSyC, Université de Technologie de Compiègne, Centre de Recherches de Royallieu, BP20529, 60205 Compiègne Cedex, France (Marc.Chardon@utc.fr, Aziz.Moukrim@utc.fr).

h, there exists an $O(n^{h(m-1)+1})$ algorithm for finding an optimal schedule. With a constant breadth m, the scheduling problem of an arbitrary precedence graph is still open. In this paper we characterize a new order class that properly contains quasi-interval orders and we prove that the Coffman–Graham (CG) algorithm yields optimal schedules for this new order class on any number of machines. This increases our knowledge about the precise location of the borderline between polynomial solvable and NP-complete problems for task systems wider than interval orders.

The organization of the paper is as follows. In the next section we give some notation and definitions. In section 3 we recall the definition of the most successors first (MSF) schedules and some results related to interval orders and quasi-interval orders. In section 4 we describe the CG algorithm and show that it is optimal for quasi-interval orders. In section 5 we introduce a larger order class, called overinterval orders, and show that the overinterval order recognition can be done in $O(n^3)$ time. In section 6, we prove that the CG algorithm is optimal for the class of overinterval orders. Finally, we discuss some extensions in the last section.

2. Notation and definitions. Let $G = (X, \prec)$ be a precedence graph. If there is a directed path of one or more arcs from task x to task y, then x is a *predecessor* of y and y is a *successor* of x. $\Gamma^+(x)$ (resp., $\Gamma^-(x)$) denotes the set of all successors (resp., predecessors) of x. Two nodes x and y are incomparable, denoted by $x||y$, if neither precedes the other. Otherwise they are said to be comparable. A subset $V \subset X$ is *linearly ordered* if V does not contain any incomparable tasks. The *transitive closure* of $G = (X, \prec)$ is denoted by $\overline{G} = (X, \overline{\prec})$. The graph, edges of which are exactly the incomparable pairs of $\prec$ on X, is called the *incomparability* graph and is denoted by G^c. A task y is an *immediate successor* (resp., *immediate predecessor*) of a task x iff y is a successor (resp., predecessor) of x and there is no task z with $x \overline{\prec} z \overline{\prec} y$ (resp., $y \overline{\prec} z \overline{\prec} x$). The *height* of a task x, denoted by $h(x)$, is the length of the longest path that starts at x. If a task has no successor, then $h(x) = 0$ and x is said to be *terminal.* A task x is *initial* if $\Gamma^-(x) = \emptyset$. We denote by $h(G)$ the height of G, which is the length of the longest path in G. Any precedence graph G can be partitioned into *levels* $i, i = h(G), \ldots, 1$: level i contains all tasks x that start paths of length $i - 1$ but not i. A *profile* is a sequence of nonnegative integers specifying the number of identical processors that are available in each time slot of length one. We shall interpret profile $M = (m_1, \ldots, m_d)$, where d is its length, to mean that each slot i in $[0, d)$ there are m_i processors available. A profile M is called *straight* if $\forall i, j \; m_i = m_j$. Otherwise, it is called *variable.* The breadth of M is denoted m and is defined as $m = \max_i m_i$.

A schedule S assigns a starting time $S(x)$ to each task x such that

1. $\forall i \; |\{x \in X \text{ such that } S(x) = i - 1\}| \leq m_i$,
2. $\forall (x, y) \in X \times X$ with $x \prec y$, $S(x) + 1 \leq S(y)$.

Our goal is to find a schedule with a minimal length.

A schedule is *full* if at each time slot before the last one all the available processors are busy. Note that any full schedule is necessarily optimal.

3. MSF schedules. The MSF algorithm is a *list scheduling* algorithm [7, 8]. The MSF algorithm first orders the tasks in a priority list L_{MSF} in nonincreasing order of their successor number. Then at each time slot, the first available processor executes the first available task in L_{MSF}.

A partial order $(X, \prec)$ is called an interval order iff its vertices $i \in X$ can be represented by intervals $[a_i, b_i[$ on the real line such that $i \prec j$ iff $b_i \leq a_j$. Also, a partial order $(X, \prec)$ is an interval order iff its transitive closure does not contain a suborder isomorphic to the structure described in Figure 1 (see [15]). Then, a partial

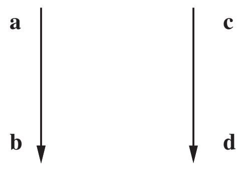
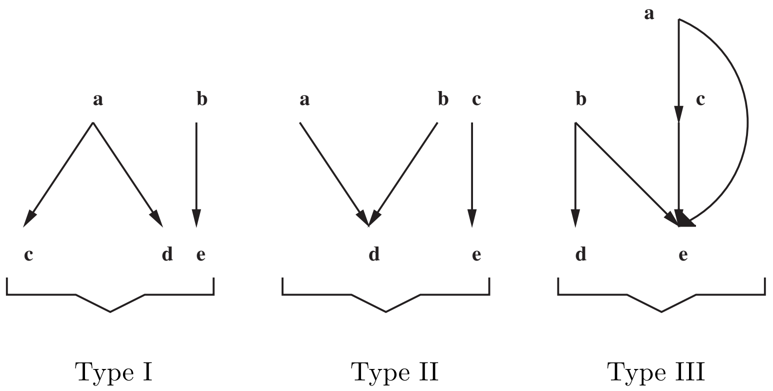


FIG. 1. *Forbidden structure for interval orders.*



Type I Type II Type III

FIG. 2. *Forbidden structures for quasi-interval orders.*

order is an interval order iff $\forall i, j \in X$, $\Gamma^+(i) \subset \Gamma^+(j)$ or $\Gamma^+(j) \subset \Gamma^+(i)$.

A partial order $(X, \prec)$ is called a *quasi-interval order* (see [13]) iff its transitive closure does not contain a suborder isomorphic to a structure of either type I, II, or III, as described in Figure 2. We can state the following lemmas.

LEMMA 1. *Let $G = (X, \prec)$ be a precedence graph. Then the following statements are evidently equivalent:*

- $\overline{G}$ *does not contain a substructure of type* I *from Figure* 2.
- *For each* 4*-tuple $a, b, c, d \in X$, with $a\overline{\prec}c, b\overline{\prec}d, a||b, a||d, c||b, c||d$, we have that if $e \in X - \{a, b, c, d\}, a\overline{\prec}e, e||b$, and $e||d$, then c and e are comparable.*
- *For each pair $i, j \in X$, with both $\Gamma^+(i) - \Gamma^+(j)$ and $\Gamma^+(j) - \Gamma^+(i)$ nonempty, we have that $\Gamma^+(i) - \Gamma^+(j)$ is linearly ordered.*

LEMMA 2. *Let $G = (X, \prec)$ be a precedence graph. Then the following statements are evidently equivalent:*

- $\overline{G}$ *does not contain a substructure of type* II *from Figure* 2.
- *For each* 4*-tuple $a, b, c, d \in X$, with $a\overline{\prec}c, b\overline{\prec}d, a||b, a||d, c||b, c||d$, we have that if $e \in X - \{a, b, c, d\}, e\overline{\prec}c, e||a$ (which implies $e \notin \Gamma^+(b)$, $e \notin \Gamma^+(d)$), then $e\overline{\prec}d$.*
- *For each pair $i, j \in X$, with both $\Gamma^+(i) - \Gamma^+(j)$ and $\Gamma^+(j) - \Gamma^+(i)$ nonempty, we have that for each $x \in \Gamma^+(i) - \Gamma^+(j)$ and each $z \in \Gamma^+(j) - \Gamma^+(i)$, if $y \in \Gamma^-(x)$ and $y||i$, then $y \in \Gamma^-(z)$.*

LEMMA 3. *Let $G = (X, \prec)$ be a precedence graph. Then the following statements are evidently equivalent:*

- $\overline{G}$ *does not contain a substructure of type* III *from Figure* 2.
- *For each* 4*-tuple $a, b, c, d \in X$, with $a\overline{\prec}c, b\overline{\prec}d, a||b, a||d, c||b, c||d$, we have that if $e \in X - \{a, b, c, d\}, b\overline{\prec}e, c\overline{\prec}e$ (which implies $d \notin \Gamma^+(e)$), then $d\overline{\prec}e$.*

- *For each pair $i, j \in X$, with both $\Gamma^+(i) - \Gamma^+(j)$ and $\Gamma^+(j) - \Gamma^+(i)$ nonempty, we have that for each $x \in \Gamma^+(i) - \Gamma^+(j)$ and each $z \in \Gamma^+(j) - \Gamma^+(i)$, if $y \in \Gamma^+(x) \cap \Gamma^+(j)$, then $y \in \Gamma^+(z)$.*

The following characterization of quasi-interval orders is straightforward from Lemmas 1, 2, and 3.

PROPOSITION 1. *Let $G = (X, \prec)$ be a precedence graph. Then the following statements are equivalent:*

(i) *G is a quasi-interval order.*

(ii) *For each pair $i, j \in X$, with both $\Gamma^+(i) - \Gamma^+(j)$ and $\Gamma^+(j) - \Gamma^+(i)$ nonempty, we have that*

- *$\Gamma^+(i) - \Gamma^+(j)$ is linearly ordered*;
- *for each $x \in \Gamma^+(i) - \Gamma^+(j)$ and each $z \in \Gamma^+(j) - \Gamma^+(i)$, if $y \in \Gamma^-(x)$ and $y||i$, then $y \in \Gamma^-(z)$;*
- *for each $x \in \Gamma^+(i) - \Gamma^+(j)$ and each $z \in \Gamma^+(j) - \Gamma^+(i)$, if $y \in \Gamma^+(x) \cap \Gamma^+(j)$, then $y \in \Gamma^+(z)$.*

The MSF algorithm optimally solves interval order problems (see [15, 12]). The same result holds for quasi-interval orders (see [13]).

4. CG schedules.

The CG algorithm (see [3, 11]) is a list scheduling algorithm, where the priority list is obtained as follows. Its construction is based on the labels $\alpha(i)$ assigned to the tasks i:

- Choose an arbitrary task x without successors and define $\alpha(x) = 1$.
- Suppose, for some $j \leq n$, that labels $1, \dots, j-1$ have been assigned. For each task x, all of whose immediate successors have already been labeled, we form a decreasing sequence using the labels of the immediate successors of x. The smallest such sequence (lexicographically) determines the task to be assigned the label j. Then the list L_{CG} used in the list scheduling is constructed in a decreasing order of the labels of the tasks. Note that no two tasks get the same label.

As mentioned before, the MSF algorithm solves quasi-interval order problems optimally. Now, we will show that the CG algorithm solves quasi-interval order problems optimally as well. First, let us note that in the general case, neither schedule dominates the other, as shown in Figure 3. For the precedence graph described in Figure 3, we have $L_{MSF} = (5, 7, 6, 4, 3, 2, 1)$ and $L_{CG} = (7, 6, 5, 4, 3, 2, 1)$. Note that in Figure 3, any task i is indexed by its CG label, $\alpha(i)$. The main result of this section is based on the following two lemmas.

LEMMA 4. *Let $G = (X, \prec)$ be a precedence graph. Then $\forall i, j \in X$, if $h(i) > h(j)$, then $\alpha(i) > \alpha(j)$.*

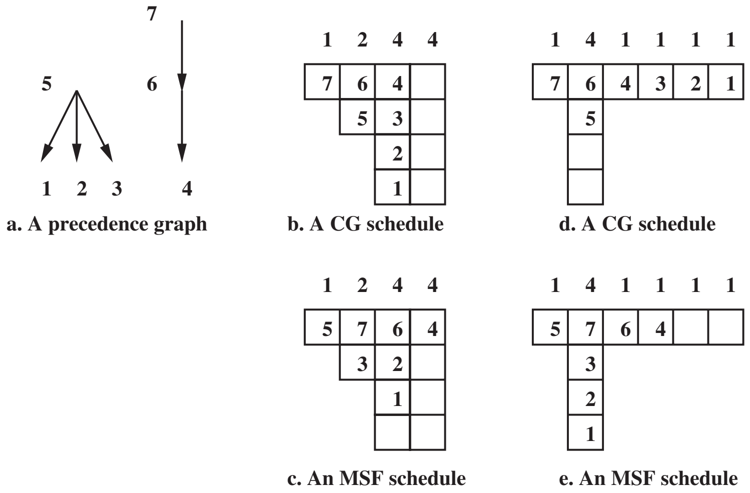
Proof. The proof is by induction on the levels of G. It suffices to note that the labels of all the tasks of any level $l = h(G) - 1, \dots, 1$ are assigned before labeling any task on level $l + 1$. □

LEMMA 5. *Let $G = (X, \prec)$ be a quasi-interval order. Then $\forall i, j \in X$, if $|\Gamma^+(i)| > |\Gamma^+(j)|$, then $\alpha(i) > \alpha(j)$.*

Proof. Let $G = (X, \prec)$ be a quasi-interval order, and let $i, j \in X$ such that $|\Gamma^+(i)| > |\Gamma^+(j)|$. We will consider two cases.

Case 1. $j \in \Gamma^+(i)$. Then $h(i) > h(j)$ and, according to Lemma 4, we have $\alpha(i) > \alpha(j)$.

Case 2. $i||j$. According to Lemma 1 and, since $|\Gamma^+(i)| > |\Gamma^+(j)|$, we consider two subcases.



a. A precedence graph

b. A CG schedule

d. A CG schedule

c. An MSF schedule

e. An MSF schedule

FIG. 3. *CG and MSF schedules: for general precedence graphs neither dominates the other.*

Case 2.1. $\Gamma^+(j) \subset \Gamma^+(i)$. So using the definition of the CG labels, we get $\alpha(i) > \alpha(j)$.

Case 2.2. $\Gamma^+(i) - \Gamma^+(j)$ and $\Gamma^+(j) - \Gamma^+(i)$ are nonempty and linearly ordered. So since $|\Gamma^+(i)| > |\Gamma^+(j)|$, we have $\Gamma^+(i) - \Gamma^+(j) = \{i_1 \dots i_s\}$ and $\Gamma^+(j) - \Gamma^+(i) = \{j_1 \dots j_r\}$ with $i_1 \prec \cdots \prec i_s$, $j_1 \prec \cdots \prec j_r$, and $s > r$. First, we will show that $\Gamma^+(i) \cap \Gamma^+(j) \subset \Gamma^+(j_r) \cap \Gamma^+(i_s)$. Indeed, let $k \in \Gamma^+(i) \cap \Gamma^+(j)$.

- $k \in \Gamma^+(j_r)$. Otherwise, since $j_r \notin \Gamma^+(i)$, we have $k||j_r$. So we would have the following for all $q \in [1 \dots s]$:
  - $i_q \notin \Gamma^+(k)$; otherwise we would have $i_q \in \Gamma^+(j)$.
  - $k \notin \Gamma^+(i_q)$ $\forall q \in [1 \dots s]$; otherwise we would have a suborder isomorphic to a forbidden structure for quasi-interval orders of type III ($\{i, j, i_q, j_r, k\}$, $\{i \overline{\prec} i_q, i_q \overline{\prec} k, j \overline{\prec} k,\ j \overline{\prec} j_r\}$).

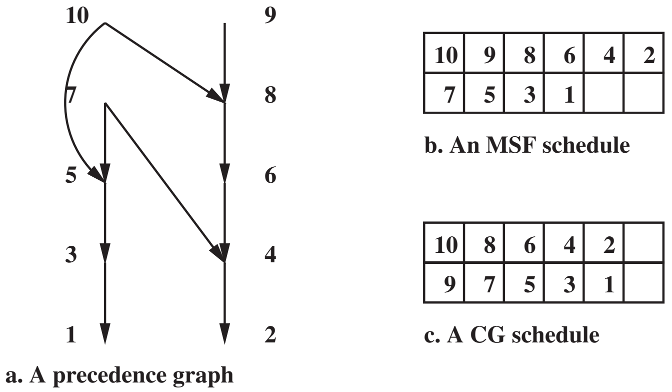
  Hence,$\forall q \in [1 \dots s]$, we have $k||i_q$. Therefore, we would have a suborder isomorphic to a forbidden structure for quasi-interval orders of type I ($\{j, i_{s-1}, k, j_r, i_s\}$, $\{j \overline{\prec} k, j \overline{\prec} j_r,\ i_{s-1} \overline{\prec} i_s\}$).
- $k \in \Gamma^+(i_s)$. Otherwise, since $i_s \notin \Gamma^+(j)$, we have $k||i_s$. So we would have a suborder isomorphic to a forbidden structure for quasi-interval orders of type III ($\{j, i, j_r, i_s, k\}$, $\{j \overline{\prec} j_r, j_r \overline{\prec} k, i \overline{\prec} k, i \overline{\prec} i_s\}$).

Now $\Gamma^+(i) \cap \Gamma^+(j) \subset \Gamma^+(j_r) \cap \Gamma^+(i_s)$ and $s > r$ imply that $h(i) > h(j)$ and, according to Lemma 4, we have $\alpha(i) > \alpha(j)$. ◻

From Lemma 5, we deduce that for quasi-interval orders, any CG schedule is an MSF schedule. Now, as the MSF algorithm is optimal for quasi-interval orders, the CG algorithm is optimal for quasi-interval orders, too. Hence, the following theorem holds.

THEOREM 1. *The CG algorithm optimally solves quasi-interval order problems with arbitrary profiles.*

The class of quasi-interval orders is a generalization of the class of interval orders. To study the behavior of the CG algorithm for precedence graphs of a larger order class than the class of quasi-interval orders, we consider the precedence graph G, described in Figure 4. G does not contain a suborder isomorphic to the forbidden



a. A precedence graph

10	9	8	6	4	2
7	5	3	1		

b. An MSF schedule

10	8	6	4	2	
9	7	5	3	1	

c. A CG schedule

Fig. 4. *MSF schedules are not optimal for overinterval orders.*

structure of type I or II for quasi-interval orders but G is not a quasi-interval order. We note that the MSF schedule is not optimal for G (see Figure 4.b) whereas the CG schedule is (see Figure 4.c).

5. A new order class (overinterval orders). In this section, we introduce a new order class, the *overinterval order* class. A partial order $G = (X, \prec)$ is an overinterval order iff its transitive closure does not contain a suborder isomorphic to a structure of type I or II described in Figure 2. Overinterval orders are a generalization of quasi-interval orders. We give a characterization of this new order class and a recognition algorithm of overinterval orders that can be implemented in $O(n^3)$. First, we state a new characterization of precedence graphs not containing a substructure of type II.

Lemma 6. *Let $G = (X, \prec)$ be a precedence graph. Then the following statements are evidently equivalent:*

- *$\overline{G}$ does not contain a substructure of type* II *from Figure* 2.
- *For each* 4*-tuple $a, b, c, d \in X$, with $a\overline{\prec}c, b\overline{\prec}d, a||b, a||d, c||b, c||d$, we have that if $e \in X - \{a, b, c, d\}, e\overline{\prec}c, e||b$, and $e||d$, then a and e are comparable.*
- *For each pair $i, j \in X$, with both $\Gamma^-(i) - \Gamma^-(j)$ and $\Gamma^-(j) - \Gamma^-(i)$ nonempty, we have that $\Gamma^-(i) - \Gamma^-(j)$ is linearly ordered.*

The following proposition is straightforward from Lemmas 1 and 6.

Proposition 2. *Let $G = (X, \prec)$ be a precedence graph. G is an overinterval order iff we have*

- *for each pair $i, j \in X$, with both $\Gamma^+(i) - \Gamma^+(j)$ and $\Gamma^+(j) - \Gamma^+(i)$ nonempty, $\Gamma^+(i) - \Gamma^+(j)$ is linearly ordered;*
- *for each pair $i, j \in X$, with both $\Gamma^-(i) - \Gamma^-(j)$ and $\Gamma^-(j) - \Gamma^-(i)$ nonempty, $\Gamma^-(i) - \Gamma^-(j)$ is linearly ordered.*

Before describing our recognition algorithm for overinterval orders and proving its correctness, we give this technical lemma.

Lemma 7. *Let $G = (X, \prec)$ be a precedence graph and $\{i_1, \ldots, i_k\} \subset X$. Then the following statements are equivalent:*

(1) *$\{i_1, \ldots, i_k\}$ is linearly ordered and $|\Gamma^+(i_1)| \geq \ldots \geq |\Gamma^+(i_k)|$.*

(2) *$i_1\overline{\prec}\cdots\overline{\prec}i_k$.*

Proof. By induction on k. □

Given that we have a characterization for overinterval orders by forbidden structures, the recognition of an overinterval order is polynomial. Indeed, it suffices to check whether all 5-element induced suborders are isomorphic to a structure of type I or II described in Figure 2. Below, we describe a recognition algorithm for overinterval orders that can be implemented to run in $O(n^3)$ time. The correctness of this algorithm follows from Proposition 2, which states that a precedence graph is not an overinterval order iff there exist $(i, j) \in X \times X$ such that at least one of the sets $\Gamma^+(i) - \Gamma^+(j)$ and $\Gamma^+(j) - \Gamma^+(i)$ is not linearly ordered (whereas they are both nonempty) or at least one of the sets $\Gamma^-(i) - \Gamma^-(j)$ and $\Gamma^-(j) - \Gamma^-(i)$ is not linearly ordered (whereas they are both nonempty).

Algorithm 1 Overinterval order recognition.

```
Require: A precedence graph $G = (X, \prec)$
Ensure: A variable FLAG which is 0 if $G$ is an overinterval order and 1 otherwise
  $FLAG := 0$;
  Compute $I := \{(i, j) \in X \times X$ such that $i || j\}$;
  while $(I \neq \emptyset)$ and $(FLAG = 0)$ do
    Let $(i, j) \in I$; $I := I - \{(i, j)\}$;
    Compute $\Gamma^+(i) - \Gamma^+(j)$, $\Gamma^+(j) - \Gamma^+(i)$, $\Gamma^-(i) - \Gamma^-(j)$, and $\Gamma^-(j) - \Gamma^-(i)$;
    if $\Gamma^+(i) - \Gamma^+(j) \neq \emptyset$ and $\Gamma^+(j) - \Gamma^+(i) \neq \emptyset$ then
      if $\Gamma^+(i) - \Gamma^+(j)$ is not linearly ordered, then $FLAG := 1$; return; end if
      if $\Gamma^+(j) - \Gamma^+(i)$ is not linearly ordered, then $FLAG := 1$; return; end if
    end if
    if $\Gamma^-(i) - \Gamma^-(j) \neq \emptyset$ and $\Gamma^-(j) - \Gamma^-(i) \neq \emptyset$, then
      if $\Gamma^-(i) - \Gamma^-(j)$ is not linearly ordered, then $FLAG := 1$; return; end if
      if $\Gamma^-(j) - \Gamma^-(i)$ is not linearly ordered, then $FLAG := 1$; return; end if
    end if
  end while.
```

Now, we will specify an appropriate data structure. The adjacency sets are stored using adjacency arrays instead of adjacency lists. For any task $i \in X$, we consider an array $a^i = (a^i_1, \dots, a^i_n)$ with $a^i_k = 1$ if $k \in \Gamma^+(i)$ and 0 otherwise. The storage requirement for this data structure can be accomplished in $O(n^2)$ time with $O(n^2)$ space. First, we preprocess the tasks by computing $|\Gamma^+(i)|$ for each task i. This can be done in $O(n^2)$ time. Then we relabel the tasks so that $|\Gamma^+(i)| \geq |\Gamma^+(i+1)|$. Now note that $k \in \Gamma^+(i) - \Gamma^+(j) = \{i_1, \dots, i_q\}$ iff $a^i_k - a^j_k = 1$. So, $\Gamma^+(i) - \Gamma^+(j)$ can be computed in $O(n)$ time for any $i, j \in X$. Moreover, according to Lemma 7, $\Gamma^+(i) - \Gamma^+(j)$ is linearly ordered iff $i_1 \overline{\prec} \cdots \overline{\prec} i_q$. This can be checked in $O(n)$ time. Now, the loop "while" repeats at most $O(n^2)$ times. This leads to the following theorem.

THEOREM 2. *The recognition of overinterval orders can be done in $O(n^3)$ time and $O(n^2)$ space.*

6. Overinterval orders and CG schedules. In this section, we will establish that CG schedules are optimal for overinterval orders and any arbitrary profile. In order to introduce this result, we need some technical lemmas.

LEMMA 8. *Let M be a profile of breadth 2 and $G = (X, \prec)$ a precedence graph, where $X = \{I_0, \dots, I_k\} \cup \{J_0, \dots, J_k\}$ is such that*

- *$(I_0, \dots, I_k)$ and $(J_0, \dots, J_k)$ are two paths of G,*
- *$\forall i \in [0, k]$ $I_i || J_i$,*
- *$\forall i \in [1, k]$ $J_{i-1} || I_i$.*

If there exists a full schedule S^* for G and M such that $\forall i \in [1,k]$, $S^*(J_i) \leq S^*(I_{i-1})$ and $S^*(J_0) < S^*(I_0)$, then the list schedule S' with the priority list $L = (I_0, J_0, \ldots I_k, J_k)$ is optimal for G and M. Moreover, we have $\forall i \in [0,k]$, $S^*(J_i) \leq S'(I_i) \leq S'(J_i) \leq S^*(I_i)$.

Proof. Let S' be the list schedule with the priority list $L = (I_0, J_0, \ldots, I_k, J_k)$ for the precedence graph G and the profile M.

Since $\forall i \in [0,k]$ we have $I_i || J_i$ and $\forall i \in [1,k]$ we have $J_{i-1} || I_i$, we can conclude that the order induced by L is a topological order (this means that if $x, y \in X$ with $x \overline{\prec} y$, then x appears before y in the list L) and $(I_0, J_0, \ldots, I_k, J_k)$ is a path in the incomparability graph G^c. It follows that, as the breadth of the profile is two, S' could be built by placing the tasks of X in the profile M according to the order specified in L. So S' is a full schedule. Now we will show that

(1) $\forall i \in [0,k]$, $S'(J_i) \leq S^*(I_i)$.

(2) $\forall i \in [0,k]$, $S^*(J_i) \leq S'(I_i)$.

Proof of (1). If $k = 0$, then $X = \{I_0, J_0\}$ and $S'(J_0) = S^*(I_0)$. If $k > 0$, then $S^*(J_1) \leq S^*(I_0)$. Therefore there exists at least one task which is executed before I_0 in S^*, whereas at most one task (I_0) is executed before J_0 in S'. So $S'(J_0) \leq S^*(I_0)$. Now, let $i \in [1,k]$. $S^*(J_i) \leq S^*(I_{i-1}) < S^*(I_i)$. Therefore at least $2i+1$ tasks ($\{J_0, \ldots, J_i\} \cup \{I_0, \ldots, I_{i-1}\}$) are executed in S^* before I_i, whereas at most $2i+1$ tasks ($\{J_0, \ldots, J_{i-1}\} \cup \{I_0, \ldots, I_i\}$) are executed in S' before J_i. Therefore $S'(J_i) \leq S^*(I_i)$.

Proof of (2). If $k = 0$, then $X = \{I_0, J_0\}$ and $S^*(J_0) = S'(I_0)$. If $k > 0$, then $S^*(J_0) < S^*(I_0)$. Therefore $S^*(J_0) = S'(I_0)$. Now, let $i \in [1,k]$. $S^*(J_i) \leq S^*(I_{i-1})$. Therefore at most $2i-1$ tasks ($\{J_0, \ldots, J_{i-1}\} \cup \{I_0, \ldots, I_{i-2}\}$) are executed in S^* before J_i, whereas at least $2i-1$ tasks ($\{J_0, \ldots, J_{i-2}\} \cup \{I_0, \ldots, I_{i-1}\}$) are executed in S' before I_i. So $S^*(J_i) \leq S'(I_i)$.

Since S' is a list schedule with the priority list $L = (I_0, J_0, \ldots, I_k, J_k)$ and the order induced by L is a *topological order* (if $x \overline{\prec} y$, then x appears before y in L), we have $S'(I_i) \leq S'(J_i)$. Hence, from (1) and (2), we have the result $S^*(J_i) \leq S'(I_i) \leq S'(J_i) \leq S^*(I_i)$. This completes the proof. □

In the rest of this section, we consider an overinterval order $G = (X, \prec)$ and an arbitrary profile M. Let I_0 and J_0 be two initial tasks of G such that $\alpha(J_0) < \alpha(I_0)$. Let S^* be a schedule of G, which fits the profile M such that $S^*(J_0) = 0$, $S^*(I_0) > 0$. We define the set $\mathcal{C}(I_0, J_0, S^*)$ as a subset of nonnegative integers in the following way: $k \in \mathcal{C}(I_0, J_0, S^*)$ iff there exist two paths $(I_0, \ldots, I_k)$ and $(J_0, \ldots, J_k)$ such that $\forall i \in [1,k]$

- $S^*(J_i) \leq S^*(I_{i-1})$;
- $\alpha(J_i) < \alpha(I_i)$;
- $S^*(T) > S^*(I_{i-1})$ $\forall T \in \Gamma^+(J_{i-1}) - \Gamma^+(J_i)$;
- $S^*(T') < S^*(J_i)$ $\forall T' \in \Gamma^-(I_i) - \Gamma^-(I_{i-1})$;
- $I_i \notin \Gamma^+(J_{i-1})$.

Note that $0 \in \mathcal{C}(I_0, J_0, S^*)$. So, $\mathcal{C}(I_0, J_0, S^*) \neq \emptyset$. Moreover, note that $\forall k \in \mathcal{C}(I_0, J_0, S^*)$, we have $k \leq |X|$, and therefore $k^* := \max\{k \geq 0, k \in \mathcal{C}(I_0, J_0, S^*)\}$ exists.

Lemma 9.

(1) $\forall i \in [1, k^*]$ *we have* $J_{i-1} || I_i$.

(2) $\forall i \in [0, k^*]$ *we have* $J_i || I_i$.

Proof. (1) Let $i \in [1, k^*]$, $I_i \notin \Gamma^+(J_{i-1})$. Moreover, $S^*(J_i) \leq S^*(I_{i-1})$ implies that $S^*(J_{i-1}) < S^*(J_i) \leq S^*(I_{i-1}) < S^*(I_i)$. So $S^*(J_{i-1}) < S^*(I_i)$, and therefore $J_{i-1} \notin \Gamma^+(I_i)$. Hence, $J_{i-1} || I_i$.

(2) As I_0 and J_0 are two initial tasks, we have $I_0||J_0$.

Let $i \in [1, k^*]$. As $\alpha(J_i) < \alpha(I_i)$, we have $I_i \notin \Gamma^+(J_i)$. Moreover, $S^*(J_i) \leq S^*(I_{i-1})$ implies that $S^*(J_i) \leq S^*(I_{i-1}) < S^*(I_i)$, and therefore $J_i \notin \Gamma^+(I_i)$. Hence, $J_i||I_i$. This completes the proof. □

LEMMA 10. *If* $\Gamma^+(J_{k^*}) \neq \emptyset$*, then* $\forall J \in \Gamma^+(J_{k^*}), S^*(J) > S^*(I_{k^*})$.

Proof. Assume that $\Gamma^+(J_{k^*}) \neq \emptyset$ and that $\{J \in \Gamma^+(J_{k^*})/S^*(J) \leq S^*(I_{k^*})\} \neq \emptyset$. So, let $U \in \Gamma^+(J_{k^*})$ such that $S^*(U) = \min\{S^*(J)/J \in \Gamma^+(J_{k^*})$ and $S^*(J) \leq S^*(I_{k^*})\}$. We will show that there exists $V \in \Gamma^+(I_{k^*}) - \Gamma^+(J_{k^*})$ such that

(i) $S^*(V) = \min\{S^*(I)/I \in \Gamma^+(I_{k^*}) - \Gamma^+(J_{k^*})$ and $\alpha(I) > \alpha(U)\}$.

(ii) $\forall T \in \Gamma^+(J_{k^*}) - \Gamma^+(U), S^*(T) > S^*(I_{k^*})$.

(iii) $\forall T' \in \Gamma^-(V) - \Gamma^-(I_{k^*}), S^*(T') < S^*(U)$.

Proof of (i). We show first that $U||I_{k^*}$.

If $U \prec I_{k^*}$, then $J_{k^*} \overline{\prec} U \prec I_{k^*}$. This contradicts the result of Lemma 9 ($J_{k^*}||I_{k^*}$). As $S^*(U) \leq S^*(I_{k^*})$, $U \notin \Gamma^+(I_{k^*})$. So $U||I_{k^*}$.

Now $\alpha(I_{k^*}) > \alpha(J_{k^*})$, $J_{k^*} \overline{\prec} U$, and $U||I_{k^*}$. So by the definition of CG labels, we have $\{I \in \Gamma^+(I_{k^*}) - \Gamma^+(J_{k^*})$ with $\alpha(I) > \alpha(U)\} \neq \emptyset$. So $\exists V \in \Gamma^+(I_{k^*}) - \Gamma^+(J_{k^*})$ such that $S^*(V) = \min\{S^*(I)/I \in \Gamma^+(I_{k^*}) - \Gamma^+(J_{k^*})$ and $\alpha(I) > \alpha(U)\}$.

Proof of (ii). Assume that there exists $T \in \Gamma^+(J_{k^*}) - \Gamma^+(U)$ such that $S^*(T) \leq S^*(I_{k^*})$. Therefore,

- $T \in \Gamma^+(J_{k^*}) - \Gamma^+(I_{k^*})$: by definition of T, $T \in \Gamma^+(J_{k^*})$. Moreover, $S^*(T) \leq S^*(I_{k^*})$ implies that $T \notin \Gamma^+(I_{k^*})$. Hence, $T \in \Gamma^+(J_{k^*}) - \Gamma^+(I_{k^*})$.
- $U \in \Gamma^+(J_{k^*}) - \Gamma^+(I_{k^*})$: by definition of U, $U \in \Gamma^+(J_{k^*})$. Moreover, $S^*(U) \leq S^*(I_{k^*})$ implies that $U \notin \Gamma^+(I_{k^*})$. Hence, $U \in \Gamma^+(J_{k^*}) - \Gamma^+(I_{k^*})$.
- $V \in \Gamma^+(I_{k^*}) - \Gamma^+(J_{k^*})$: that follows from (i).
- $T||U$: As $T \in \Gamma^+(J_{k^*}) - \Gamma^+(U)$, $T \notin \Gamma^+(U)$. Moreover, $U \notin \Gamma^+(T)$. Otherwise we would have $J_{k^*} \overline{\prec} T \prec U$, and therefore $S^*(T) < S^*(U) \leq S^*(I_{k^*})$. Then $S^*(T) < S^*(U)$ would contradict the definition of $S^*(U) = \min\{S^*(J)/J \in \Gamma^+(J_{k^*})$ and $S^*(J) \leq S(I_{k^*})\}$. Hence, $T||U$.

Therefore, $\Gamma^+(I_{k^*}) - \Gamma^+(J_{k^*}) \neq \emptyset$, $\Gamma^+(J_{k^*}) - \Gamma^+(I_{k^*}) \neq \emptyset$, and $\Gamma^+(J_{k^*}) - \Gamma^+(I_{k^*})$ is not linearly ordered. This contradicts the characterization of overinterval orders (see Proposition 2).

Proof of (iii). Assume that there exists $T' \in \Gamma^-(V) - \Gamma^-(I_{k^*})$ such that $S^*(T') \geq S^*(U)$. Therefore, as in the proof of (ii), one can verify that

- $I_{k^*} \in \Gamma^-(V) - \Gamma^-(U)$.
- $T' \in \Gamma^-(V) - \Gamma^-(U)$.
- $J_{k^*} \in \Gamma^-(U) - \Gamma^-(V)$.
- $T'||I_{k^*}$.

Therefore, $\Gamma^-(U) - \Gamma^-(V) \neq \emptyset$, $\Gamma^-(V) - \Gamma^-(U) \neq \emptyset$, and $\Gamma^-(V) - \Gamma^-(U)$ is not linearly ordered. This contradicts the characterization of overinterval orders (see Proposition 2).

From (i), (ii), and (iii), we have

- $I_{k^*} \overline{\prec} V$.
- $J_{k^*} \overline{\prec} U$.
- $S^*(U) \leq S^*(I_{k^*})$.
- $\alpha(U) < \alpha(V)$.
- $S^*(T) > S^*(I_{k^*})$ $\forall T \in \Gamma^+(J_{k^*}) - \Gamma^+(U)$.
- $S^*(T') < S^*(U)$ $\forall T' \in \Gamma^-(V) - \Gamma^-(I_{k^*})$.
- $V \notin \Gamma^+(J_{k^*})$.

It follows that if we assume $\{J \in \Gamma^+(J_{k^*})/S^*(J) \leq S^*(I_{k^*})\} \neq \emptyset$, we would have a contradiction with $k^* = \max\{k \geq 0, k \in \mathcal{C}(I_0, J_0, S^*)\}$. This completes the proof. □

Now, we give our main result.

THEOREM 3. *The CG algorithm is optimal for overinterval orders and arbitrary profiles.*

Proof. Assume that for some overinterval order $G = (X, \prec)$, the CG algorithm is not optimal, with X being as small as possible. Let S_{CG} be a CG schedule and S^* be an optimal schedule. For each time instant t, we define $O_{CG}(t)$ and $O^*(t)$ as $O_{CG}(t) = \{x \in X / S_{CG}(x) = t\}$ and $O^*(t) = \{x \in X / S^*(x) = t\}$.

We will show that S^* can be transformed into another optimal schedule S' such that $O'(0) = O_{CG}(0)$, with $O'(t) = \{x \in X / S'(x) = t\}$. This would contradict the minimality of G. First, we will show that S^* can be chosen such that $|O^*(0)| = |O_{CG}(0)|$ by considering two cases.

Case 1. $|O^*(0)| < |O_{CG}(0)|$. So, S^* has an idle processor at time $t = 0$ and there exists a task $I \in O_{CG}(0) - O^*(0)$. By executing I at time $t = 0$ in S^*, we also obtain an optimal schedule. Repeating this operation $|O_{CG}(0) - O^*(0)|$ times, we can assume that $|O^*(0)| = |O_{CG}(0)|$.

Case 2. $|O^*(0)| > |O_{CG}(0)|$. So there exists an available task I and an idle processor at time $t = 0$ in S_{CG}. This contradicts the CG algorithm scheme.

Hence, we can assume that $|O^*(0)| = |O_{CG}(0)|$. If $O^*(0) \neq O_{CG}(0)$, then there exist $I_0, J_0 \in X$ such that $I_0 || J_0, S^*(J_0) = 0, S^*(I_0) > 0, S_{CG}(I_0) = 0, S_{CG}(J_0) > 0$, and $\alpha(I_0) > \alpha(J_0)$. Now, let $k^* = \max\{k \geq 0, k \in \mathcal{C}(I_0, J_0, S^*)\}$. According to Lemma 9, we have $\forall i \in [0, k^*]$: $I_i || J_i$ and $\forall i \in [1, k]$ $J_{i-1} || I_i$.

Moreover, as $(I_0, \ldots, I_{k^*})$ and $(J_0, \ldots, J_{k^*})$ are two paths, at each time instant at most two tasks of the set $\{I_0, \ldots, I_{k^*}\} \cup \{J_0, \ldots, J_{k^*}\}$ are executed. Without considering the tasks of the set $X - (\{I_0, \ldots, I_{k^*}\} \cup \{J_0, \ldots, J_{k^*}\})$ and by proceeding as in Lemma 8, we replace the tasks of $\{I_0, \ldots, I_{k^*}\} \cup \{J_0, \ldots, J_{k^*}\}$ by using the list schedule with the priority list $L = (I_0, J_0, \ldots, I_{k^*}, J_{k^*})$. We denote the new schedule by S'. Note that it suffices to prove that S' is feasible in order to conclude that it is optimal.

According to Lemma 8, $\forall i \in [0, k^*], S^*(J_i) \leq S'(I_i) \leq S'(J_i) \leq S^*(I_i)$. So, $\forall i \in [0, k^*], S^*(J_i) \leq S'(J_i)$ and $S'(I_i) \leq S^*(I_i)$. Then, to check that S' is feasible, it suffices to show that $\forall i \in [0, k^*]$,

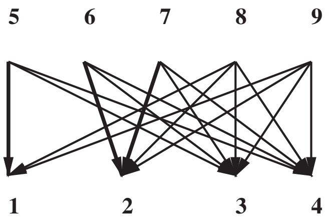
(i) $\forall T \in \Gamma^+(J_i) - (\{I_0, \ldots, I_{k^*}\} \cup \{J_0, \ldots, J_{k^*}\})$, $S'(J_i) < S'(T) (= S^*(T))$.

(ii) $\forall U \in \Gamma^-(I_i) - (\{I_0, \ldots, I_{k^*}\} \cup \{J_0, \ldots, J_{k^*}\})$, $S'(U) (= S^*(U)) < S'(I_i)$.

(i) Let $i \in [0, k^*]$ and $T \in \Gamma^+(J_i) - (\{I_0, \ldots, I_{k^*}\} \cup \{J_0, \ldots, J_{k^*}\})$. We will consider the following two cases:
- $J_{k^*} \overline{\prec} T$. So according to Lemma 10, $S^*(T) > S^*(I_{k^*})$. As $S^*(I_{k^*}) \geq S'(J_{k^*})$, $S^*(T) > S'(J_{k^*})$.
- $\exists i_0 \geq i$ such that $T \in \Gamma^+(J_{i_0}) - \Gamma^+(J_{i_0+1})$. So, according to the definition of $\mathcal{C}(I_0, J_0, S^*)$, we have $S^*(T) > S^*(I_{i_0})$. Now, according to Lemma 8, $S^*(I_{i_0}) \geq S'(J_{i_0})$. So $S^*(T) > S'(J_{i_0}) \geq S'(J_i)$ $(i \leq i_0)$.

(ii) Let $i \in [0, k^*]$ and $U \in \Gamma^-(I_i) - (\{I_0, \ldots, I_{k^*}\} \cup \{J_0, \ldots, J_{k^*}\})$. As $\Gamma^-(I_0) = \emptyset$, there exists $i_0 \leq i$ such that $U \in \Gamma^-(I_{i_0}) - \Gamma^-(I_{i_0-1})$. According to the definition of $\mathcal{C}(I_0, J_0, S^*)$, we have $S^*(U) < S^*(J_{i_0})$, and according to Lemma 8, $S^*(J_{i_0}) \leq S'(I_{i_0})$. So, $S^*(U) < S'(I_{i_0}) \leq S'(I_i)$ $(i_0 \leq i)$.

Job I_0 is now in $O^*(0)$ too, since I_0 has taken the place of J_0 in the modified optimal schedule S^*. Repeating the transformation used in the proof of Lemma 8 $|O^*(0) - O_{CG}(0)|$ times, we obtain another optimal schedule S' such that $O'(0) = O_{CG}(0)$ and have a contradiction. This completes the proof. □



a. A precedence graph

9	6	4	1
8	5	3	
7		2	

b. A CG schedule

9	7	4	
8	6	3	
5	1	2	

b. An optimal schedule

Fig. 5.

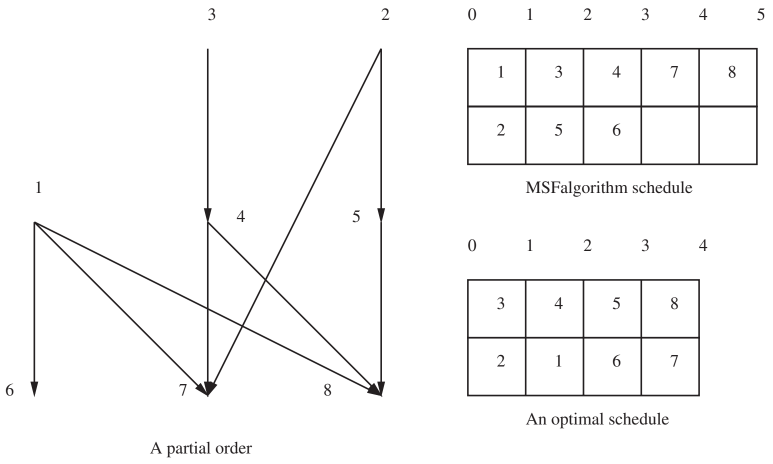
7. Discussion. In this paper, we have introduced the class of overinterval orders. Next, we have shown that the CG algorithm solves the nonpreemptive scheduling problem of UET task systems with overinterval orders on arbitrary profiles. A natural extension of this work is to study the behavior of the CG algorithm for larger order classes than overinterval orders. Unfortunately, as soon as a precedence graph contains a suborder isomorphic to a forbidden structure for overinterval orders of type I (see Figure 3) or of type II (see Figure 5), the CG algorithm does not ensure obtaining optimal schedules.

Another interesting extension is the scheduling problem of UET task systems in the presence of unit communication delays, also called UECT task systems (see [2]). In this case, a precedence constraint $i \prec j$ states that a data transfer must occur after the end of task i and before the beginning of task j. If i and j are not assigned to the same processor, a delay of length $c_{ij} = 1$ is needed to forward the data. A schedule S assigns a starting time $S(i)$ and a processor $\pi(i) \in \{1, \ldots, m\}$ to each task i such that

1. $\forall i, j \in X$, if $\pi(i) = \pi(j)$, then $S(i) + 1 \leq S(j)$ or $S(j) + 1 \leq S(i)$.
2. $\forall (i, j) \in X \times X$ with $i \prec j$, if $\pi(i) = \pi(j)$, then $S(i) + 1 \leq S(j)$ else $S(i) + 2 \leq S(j)$.

Our goal is to find a schedule that minimizes the time taken to finish all the tasks.

We have proven that the MSF algorithm optimally solves UECT task systems with quasi-interval orders (see [14]). This no longer remains true for overinterval orders. A counterexample is illustrated in Figure 6, where $L_{MSF} = (1, 2, 3, 4, 5, 6, 7, 8)$. Hanen and Munier (see [9]) have studied the behavior of the CG algorithm for the UECT problem. The task system considered in Figure 7 shows that the CG algorithm does not always lead to an optimal solution for UECT task systems with overinterval orders ($L_{CG} = (8, 7, 6, 5, 4, 3, 2, 1)$).



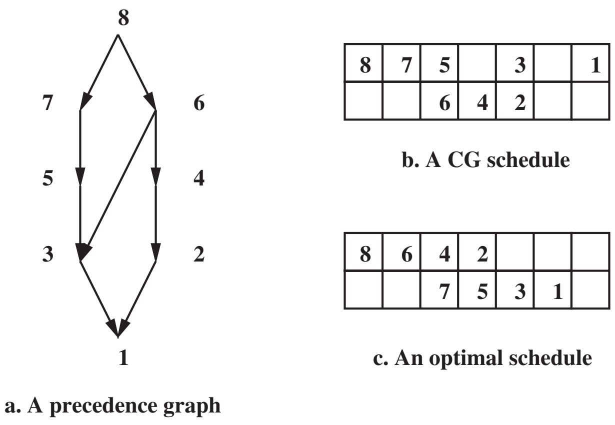
1	3	4	7	8
2	5	6		

MSFalgorithm schedule

3	4	5	8
2	1	6	7

An optimal schedule

Fig. 6. *MSF schedules are not optimal for UECT overinterval order task systems.*



8	**7**	**5**		**3**		**1**
		6	**4**	**2**		

b. A CG schedule

8	**6**	**4**	**2**			
		7	**5**	**3**	**1**	

c. An optimal schedule

Fig. 7. *CG schedules are not optimal for UECT overinterval order task systems.*

Acknowledgments. We are very grateful to the anonymous referees for helpful suggestions that enabled us to improve the paper.

REFERENCES

[1] M. Bartusch, R. H. Möhring, and F. J. Radermacher, *Some remarks on the m-machine unit time scheduling problem*, in Proceedings of the 11th International Workshop on Graph-Theoretic Concepts in Computer Science (WG'85), H. Noltemeier, ed., Trauner-Verlag, Linz, 1985, pp. 23–26.

[2] Ph. Chrétienne and C. Picouleau, *Scheduling with communication delays: A survey*, in Scheduling Theory and Its Applications, P. Chrétienne, E. G. Coffman, J. K. Lenstra, and Z. Liu, eds., John Wiley, Chichester, 1995, pp. 65–90.

[3] E. G. Coffman, Jr. and R. L. Graham, *Optimal scheduling for two-processor systems*, Acta Informat., 1, (1972), pp. 200–213.

[4] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, 2nd ed., MIT Press, Cambridge, MA, 2001.

[5] D. Dolev and M. K. Warmuth, *Scheduling precedence graphs of bounded height*, J. Algorithms, 5 (1984), pp. 48–59.

[6] D. Dolev and M. K. Warmuth, *Profile scheduling of opposing forests and level orders*, SIAM J. Alg. Disc. Meth., 6 (1985), pp. 665–687.

[7] R. L. Graham, *Bounds on multiprocessing anomalies*, Bell System Tech. J., 45 (1966), pp. 1563–1581.

[8] R. L. Graham, *Bounds on multiprocessing timing anomalies*, SIAM J. Appl. Math., 17 (1969), pp. 416–429.

[9] C. Hanen and A. Munier, *Performance of Coffman–Graham schedules in the presence of unit communication delays*, Discrete Appl. Math., 81 (1998), pp. 93–108.

[10] T. C. Hu, *Parallel sequencing and assembly line problems*, Oper. Res., 9 (1961), pp. 841–848.

[11] S. Lam and R. Sethi, *Worst case analysis of two scheduling algorithms*, SIAM J. Comput., 6 (1977), pp. 518–536.

[12] Z. Liu and E. Sanlaville, *Profile scheduling by list algorithms*, in Scheduling Theory and Its Applications, P. Chrétienne, E. G. Coffman, J. K. Lenstra, and Z. Liu, eds., John Wiley, Chichester, 1995, pp. 91–110.

[13] A. Moukrim, *Optimal scheduling on parallel machines for a new order class*, Oper. Res. Lett., 24 (1999), pp. 91–95.

[14] A. Moukrim, *Scheduling unitary task systems with zero-one communication delays for quasi-interval orders*, Discrete Appl. Math., 127 (2003), pp. 461–476.

[15] C. H. Papadimitriou and M. Yannakakis, *Scheduling interval-ordered tasks*, SIAM J. Comput., 8 (1979), pp. 405–409.

[16] H. Pelzer, *Zur Komplexität des* 3*-Maschinen-Scheduling problems*, diploma thesis, RWTH Aachen, University, Aachen, Germany, 1984.

[17] J. D. Ullman, *NP-complete scheduling problems*, J. Comput. Systems Sci., 10 (1975), pp. 384–393.

SIAM J. DISCRETE MATH.
Vol. 19, No. 1, pp. 122–134

TIGHTER BOUNDS FOR GRAPH STEINER TREE APPROXIMATION*

GABRIEL ROBINS† AND ALEXANDER ZELIKOVSKY‡

Abstract. The classical Steiner tree problem in weighted graphs seeks a minimum weight connected subgraph containing a given subset of the vertices (terminals). We present a new polynomial-time heuristic that achieves a best-known approximation ratio of $1 + \frac{\ln 3}{2} \approx 1.55$ for general graphs and best-known approximation ratios of ≈ 1.28 for both quasi-bipartite graphs (i.e., where no two nonterminals are adjacent) and complete graphs with edge weights 1 and 2. Our method is considerably simpler and easier to implement than previous approaches. We also prove the first known nontrivial performance bound ($1.5 \cdot \mathrm{OPT}$) for the iterated 1-Steiner heuristic of Kahng and Robins in quasi-bipartite graphs.

Key words. Steiner trees, approximation algorithms, graph Steiner problem, iterated 1-Steiner

AMS subject classifications. 05C05, 05C85

DOI. 10.1137/S0895480101393155

1. Introduction.

Given an arbitrary weighted graph with a distinguished vertex subset, the *Steiner tree problem* seeks a minimum-cost subtree spanning the distinguished vertices. Steiner trees are important in various applications such as VLSI routing [14], wirelength estimation [7], phylogenetic tree reconstruction in biology [11], and network routing [12]. The Steiner tree problem is NP-hard, even in the Euclidean or rectilinear metrics [8], and thus efficient approximation heuristics are sought instead of exact algorithms.

Arora established that Euclidean and rectilinear minimum-cost Steiner trees can be efficiently approximated arbitrarily close to optimal [2]. On the other hand, unless $P = NP$, the Steiner tree problem in general graphs cannot be approximated within a factor of $1 + \epsilon$ for sufficiently small $\epsilon > 0$ [5]. For arbitrary weighted graphs, the best Steiner approximation ratio achievable within polynomial time was gradually improved from 2 to 1.59 in a series of papers [21, 22, 3, 23, 18, 15, 10].

In this paper we address the graph Steiner tree problem by presenting a polynomial-time approximation scheme with a best-known performance ratio approaching $1 + \frac{\ln 3}{2} \approx 1.55$. This improves upon the previously best-known ratio of 1.59 due to Hougardy and Prömel [10]. We apply our heuristic for the Steiner tree problem to quasi-bipartite graphs (i.e., graphs in which no two nonterminals are adjacent), where our heuristic achieves an approximation ratio of ≈ 1.28 within time $O(mn^2)$ (m and n are the number of terminals and nonterminals in the graph, respectively). This is an improvement over the primal-dual algorithm of Rajagopalan and Vazirani [19], where the bound exceeds 1.5.

*Received by the editors July 31, 2001; accepted for publication (in revised form) November 3, 2004; published electronically June 16, 2005. A preliminary version of this work appeared in [20].
http://www.siam.org/journals/sidma/19-1/39315.html

†Department of Computer Science, University of Virginia, Charlottesville, VA 22904 (robins@cs.virginia.edu, http://www.cs.virginia.edu/~robins). The work of this author was supported in part by a Packard Foundation Fellowship, by National Science Foundation Young Investigator Award MIP-9457412, and by NSF grant CCR-9988331.

‡Department of Computer Science, Georgia State University, Atlanta, GA 30303 (alexz@cs.gsu.edu, http://www.cs.gsu.edu/~cscazz). The work of this author was supported in part by NSF grant CCR-9988331 and by a GSU Research Initiation Grant.

We also show that the well-known iterated 1-Steiner heuristic of Kahng and Robins [13, 9, 14] achieves an approximation ratio of 1.5 in quasi-bipartite graphs. Previously, no nontrivial bounds were known for the iterated 1-Steiner heuristic. Finally, we improve the approximation ratio achievable for the Steiner tree problem in complete graphs with edge weights 1 and 2 from the previously best-known bound of $\frac{4}{3}$ [5] to less than 1.28 for our algorithm.

The remainder of this paper is organized as follows. In the next section we introduce basic definitions, notation, and properties. In section 3 we present our main algorithm, the k-restricted loss-contracting algorithm (k-LCA). The basic approximation result for the k-LCA is proved in section 4. In sections 5 and 6 we prove an approximation ratio of the k-LCA in general graphs and estimate the performance of the iterated 1-Steiner and the k-LCA heuristics in both quasi-bipartite graphs and complete graphs with weights 1 and 2. We conclude in section 7 with possible future research directions.

2. Definitions, notation, and basic properties. Let $G = (V, E, cost)$ be a graph with nonnegative edge costs. Any tree in G spanning a given set of *terminals* $S \subseteq V$ is called a *Steiner tree*, and the cost of a tree is defined to be the sum of its edge costs. The *Steiner tree problem* seeks a minimum-cost Steiner tree for a given terminal set S. Any nonterminal vertices contained in a Steiner tree are referred to as *Steiner points*. We can assume that the graph edge cost function is metric (i.e., the triangle inequality holds) since we can replace any edge $e \in E$ with the shortest path connecting the ends of e. Henceforth, we will therefore assume that G is a complete graph. Similarly, for the subgraph G_S induced by the terminal set S, let G_S be the complete graph with vertex set S.

Let $MST(G_S)$ be a minimum spanning tree of G_S. For any graph H, let $cost(H)$ be the sum of the costs of all edges in H. We thus denote the cost of a minimum spanning tree of H by $mst(H)$, e.g., $cost(MST(G_S)) = mst(G_S)$. For brevity, we use mst to denote $mst(G_S)$.

A Steiner tree over a terminal subset $S' \subset S$ in which all terminals S' are leaves is called a *full component* (see Figure 1(a)). Any Steiner tree can be decomposed into full components by splitting all the nonleaf terminals. Our algorithm will proceed by adding full components to a growing solution, based on their "relative cost savings" (this notion will be made precise below). We assume that any full component has its own copy of each Steiner point so that full components chosen by our algorithm do not share Steiner points.

A Steiner tree that does not contain any Steiner points (i.e., where each full component consists of a single edge) will henceforth be called a *terminal-spanning tree*. Our algorithm will compute relative cost savings with respect to a "shrinking" terminal-spanning tree, which initially coincides with $MST(G_S)$.

The relative cost saving of a full component is quantified by the ratio of how much that full component decreases the cost of the current terminal-spanning tree over the cost of connecting its Steiner points to terminals. The cost savings of an arbitrary graph H with respect to a terminal-spanning tree T is the difference between the cost of T and the cost of the Steiner tree in the graph obtained by augmenting H with the tree T. Let $T[H]$ be the minimum-cost graph in $H \cup T$, which contains H and spans all the terminals of S (see Figure 2). The *gain* of H with respect to T is defined as $gain_T(H) = cost(T) - cost(T[H])$. If H is a Steiner tree, then $gain_T(H) = cost(T) - cost(H)$. Note that $gain_T(H) \leq cost(T) - mst(T \cup H)$ since $T[H]$ cannot cost less than $MST(T \cup H)$. In fact, the gain of a full component K

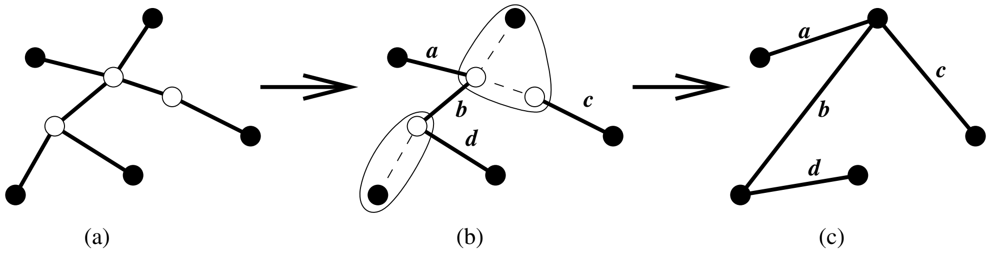


FIG. 1. (a) *A full component K: filled circles denote terminals and hollow circles denote Steiner points.* (b) *Connected components of $Loss(K)$ to be collapsed; dashed edges belong to $Loss(K)$.* (c) *The corresponding terminal-spanning tree $C[K]$ with the contracted $Loss(K)$.*

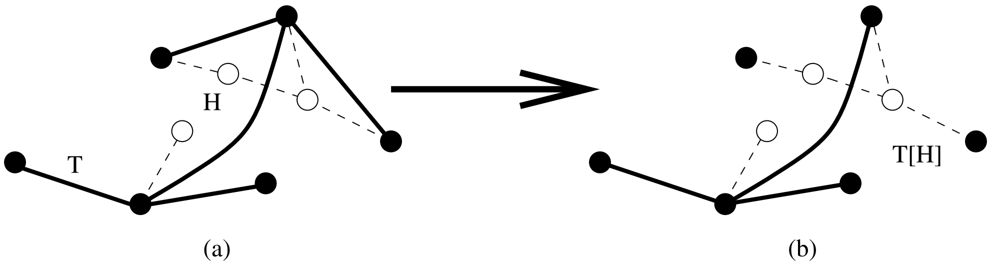


FIG. 2. (a) *A graph H (dashed edges) and a terminal-spanning tree T (solid edges).* (b) *The corresponding graph $T[H]$ contains H and spans all of the terminals.*

also can be defined as

$$gain_T(K) = cost(T) - mst(T \cup E_0(K)) - cost(K),$$

where $E_0(K)$ are zero-cost edges between all pairs of terminals of K. For brevity, the minimum spanning tree of $T \cup E_0$ will be referred to as T/E_0 for any set of zero-cost edges between pairs of terminals in S.

We will use the following property of *gain* (see Lemma 3.3-4, p. 465 in [22] and Lemma 3.14, p. 391 in [3]). Let E_0 be an arbitrary set of zero-cost edges between pairs of terminals, and let K be a full component. Then

$$gain_{T/E_0}(K) \leq gain_T(K).$$

This property implies the following key property of *gain*.

LEMMA 2.1. *For any terminal-spanning tree T and full components $K_1, K_2, \ldots, K_n$,*

$$gain_T\left(\bigcup_{i=1}^{n} K_i\right) \leq \sum_{i=1}^{n} gain_T(K_i).$$

Proof. The proof follows from the following chain of inequalities:

$$\begin{aligned} gain_T\left(\bigcup_{i=1}^{n} K_i\right) &= cost(T) - cost\left(T/\bigcup_{i=1}^{n} E_0(K_i)\right) - \sum_{i=1}^{n} cost(K_i) \\ &= cost(T) - cost(T \cup E_0(K_1)) - cost(K_1) \\ &\qquad + cost(T/E_0(K_1)) - cost(T/E_0(K_1) \cup E_0(K_2)) - cost(K_2) \\ &\quad \cdots \end{aligned}$$

$$+ \mathit{cost}\left(T/\bigcup_{i=1}^{n-1} E_0(K_i)\right) - \mathit{cost}\left(T/\bigcup_{i=1}^{n} E_0(K_i)\right) - \mathit{cost}(K_n)$$

$$= \sum_{i=1}^{n} \mathit{gain}_{T/\bigcup_{j\le i-1} E_0(K_j)}(K_i)$$

$$\le \sum_{i=1}^{n} \mathit{gain}_T(K_i). \qquad \square$$

The minimum-cost connection of the Steiner points of a full component K to its terminals is denoted $\mathit{Loss}(K)$. Formally, $\mathit{Loss}(K)$ is a minimum-cost subgraph of K containing a path from each Steiner point of K to one of the terminals of K (see Figure 1(b)). The following lemma gives a simple method of computing $\mathit{Loss}(K)$.

Lemma 2.2. *For any full component K, $\mathit{Loss}(K) = MST(K \cup E_0) \setminus E_0$, where $K \cup E_0$ is K combined with zero-cost edges E_0 between all pairs of terminals in K.*

Proof. The forest $F = MST(K \cup E_0) \setminus E_0$ connects all Steiner points of K to the terminals of K and has cost $MST(K \cup E_0)$. Note that F has the minimum possible cost since $\mathit{Loss}(K) \cup E_0$ spans all the vertices of K and therefore cannot cost more than $MST(K \cup E_0)$. $\square$

Intuitively, *Loss* will serve as an upper bound on the optimal solution cost increase during our algorithm's execution (as will be elaborated below). We will denote the cost of $\mathit{Loss}(K)$ by $\mathit{loss}(K)$. The loss of a union of full components is the sum of their individual losses.

As soon as our algorithm selects a full component K it *contracts* its $\mathit{Loss}(K)$, i.e., "collapses" each connected component of $\mathit{Loss}(K)$ into a single node (see Figure 1(c)). Formally, a *loss-contracted* full component $C[K]$ is a terminal-spanning tree over the terminals of K in which two terminals are connected if there is an edge between the corresponding two connected components in the forest $\mathit{Loss}(K)$. The cost of any edge in $C[K]$ coincides with the cost of the corresponding edge in K. The 1-to-1 correspondence between edges of $K \setminus \mathit{Loss}(K)$ and $C[K]$ implies that $\mathit{cost}(H) - \mathit{loss}(H) = \mathit{cost}(C[H])$. Similarly, for any Steiner tree H, $C[H]$ is the terminal-spanning tree in which the losses of all full components of H are contracted.

Our algorithm constructs a *k-restricted* Steiner tree, i.e., a Steiner tree in which each full component has at most k terminals. Let Opt_k be an optimal k-restricted Steiner tree, and let opt_k and loss_k be the cost and loss of Opt_k, respectively. Let *opt* and *loss* be the cost and loss of the optimal Steiner tree, respectively.

The following lemma shows that if no k-restricted full component can improve a Steiner tree H, then H cannot be very expensive; i.e., if we contract the loss of each full component of H, then the resulting tree costs no more than an optimal k-restricted Steiner tree.

Lemma 2.3. *Let H be a Steiner tree; if $\mathit{gain}_{C[H]}(K) \le 0$ for any k-restricted full component K, then*

$$\mathit{cost}(H) - \mathit{loss}(H) = \mathit{cost}(C[H]) \le \mathit{opt}_k.$$

Proof. Let $K_1, \ldots, K_p$ be full components of Opt_k. The proof follows from the following chain of inequalities:

$$\begin{aligned} \mathit{cost}(C[H]) - \mathit{opt}_k &= \mathit{gain}_{C[H]}(\mathit{Opt}_k) \\ &= \mathit{gain}_{C[H]}(K_1 \cup \cdots \cup K_p) \\ &\le \mathit{gain}_{C[H]}(K_1) + \cdots + \mathit{gain}_{C[H]}(K_p) \\ &\le 0. \qquad \square \end{aligned}$$

Input: A complete graph $G = (V, E, cost)$ with edge costs satisfying the triangle inequality, a set of terminals $S \subseteq V$, and an integer k, $3 \leq k \leq |S|$
Output: A k-restricted Steiner tree in G connecting all the terminals in S

```
$T = MST(G_S)$
$H = G_S$
Repeat forever
    Find a $k$-restricted full component $K$ with at least 3 terminals
        maximizing $r = gain_T(K)/loss(K)$
    If $r \leq 0$ then exit repeat
    $H = H \cup K$
    $T = MST(T \cup C[K])$
Output the tree $MST(H)$
```

Fig. 3. *The k-LCA.*

An *approximation ratio* of an algorithm is an upper bound on the ratio of the cost of the found solution over the cost of an optimal solution. In the next section we will propose a new algorithm for the Steiner tree problem and prove a (best-to-date) approximation ratio for it.

3. The algorithm. All previous heuristics (except those of the Berman–Ramaiyer [3] approach) with provably good approximation ratios repeatedly choose appropriate full components and then contract them to form the overall solution. However, this strategy does not allow us to discard an already accepted full component even if later we would find out that a better full component *conflicts* with a previously accepted component (two components conflict if they share at least two terminals).

The main idea behind the loss-contracting algorithm (see Figure 3) is to contract as little as possible so that (i) a chosen full component may still participate in the overall solution, but (ii) not many other full components would be rejected. In particular, if we contract $Loss(K)$, i.e., replace a full component K with $C[K]$, then (i) it will not cost anything to add a full component K to the overall solution, and (ii) we decrease the gain of full components, which conflict with K by a small value (e.g., less than in the Berman–Ramaiyer algorithm for large k and much smaller than in [15] for any k).

Our algorithm iteratively modifies a terminal-spanning tree T, which is initially $MST(G_S)$, by incorporating into T loss-contracted full components greedily chosen from G. Each such component K has positive gain, and therefore contains at least three terminals and has nonzero loss. The intuition behind the gain-over-loss objective ratio is as follows. The cost of the approximate solution lies between $mst = mst(G_S)$ and opt_k. If we accept a component K, then it increases (by the gain of K) the gap between mst and the cost of the approximation. Thus the gain of K is our clear profit. On the other hand, if K does not belong to Opt_k, then after accepting K we would no longer be able to reach Opt_k because we would need to compensate for the connection of incorrectly chosen Steiner points. Therefore, the value of $loss(K)$, which is the connection cost of Steiner points of K to terminals, is an upper bound on the increase in the cost gap between opt_k and the best achievable solution after accepting K. Thus $loss(K)$ is an estimate of our connection expense. Maximizing the ratio of gain over loss is equivalent to maximizing the profit per unit expense.

We now describe a polynomial-time implementation of the k-LCA. We first find all pairwise distances in the graph G. Then, for each k-tuple of terminals (there are $|S|^k$ of them) it is sufficient to try all possible choices of $k-2$ Steiner points chosen from the nonterminal nodes of $V-S$ because every k-restricted full component K is uniquely defined by its Steiner points of degree at least 3. The loss of K can be determined in time $O(k^2)$ by finding the minimum spanning tree of $K \cup E_0$ (see Lemma 2.2). Thus, we can find all full Steiner components in time $O(|S|^k \cdot |V-S|^{k-2})$. Note that the cost and loss do not change in the iterations of the k-LCA.

The number of iterations of k-LCA cannot exceed the number of full Steiner components $O(S^k)$ since we have $gain_T(K) = 0$ after contracting the loss of a full component K. The gain of a full component K can be found in time $O(k)$ after precomputing the longest edges between any pair of nodes in the current minimum spanning tree, which may be accomplished in time $O(S \log S)$. Thus, the runtime of all the iterations can be bounded by $O(k \cdot S^{2k+1} \log S)$. The total runtime is thus $O(|S|^k \cdot |V-S|^{k-2} + k \cdot S^{2k+1} \log S)$.

4. Approximation ratio of the *k*-LCA.

This section proves the basic approximation result of this paper.

THEOREM 4.1. *For any instance of the Steiner tree problem, the cost of the approximate Steiner tree produced by the k-LCA is at most*

$$Approx \le loss_k \cdot \ln\left(1 + \frac{mst - opt_k}{loss_k}\right) + opt_k. \tag{4.1}$$

Proof. Let $K_1, \ldots, K_{last}$ be full components chosen by the k-LCA. Let $T_0 = MST(G_S)$ and let T_i, $i = 1, \ldots, last$ be the tree T produced by the k-LCA after i iterations. Let $cost(T_i)$ be the cost of T_i after the ith iteration of the k-LCA.

LEMMA 4.2. $gain_{T_{i-1}}(K_i) = cost(T_{i-1}) - mst(T_{i-1} \cup K_i)$.

Proof. It is sufficient to show that $T_{i-1}[K_i] = MST(T_{i-1} \cup K_i)$. Assume that $MST(T_{i-1} \cup K_i)$ does not contain some edge $e \in K_i$ and let A and B be two connected components of $K_i - \{e\}$. We will show that either A or B has a larger gain-over-loss ratio, which contradicts the choice of K_i.

Since e does not belong to $MST(T_{i-1} \cup K_i)$, we have $cost(T_{i-1}[A \cup B]) < cost(T_{i-1}[K_i])$. By Lemma 2.1, $gain_{T_{i-1}}(K_i) < gain_{T_{i-1}}(A \cup B) \le gain_{T_{i-1}}(A) + gain_{T_{i-1}}(B)$. Since $e \notin MST(T_{i-1} \cup K_i)$, we conclude that $e \notin MST(K_i \cup E_0)$, where E_0 are zero-cost edges between all terminals of K_i, and by Lemma 2.2, $e \notin Loss(K_i)$. Thus $Loss(K_i) = Loss(A) \cup Loss(B)$ and $loss(K_i) = loss(A) + loss(B)$. Finally,

$$\frac{gain_{T_{i-1}}(K_i)}{loss(K_i)} < \frac{gain_{T_{i-1}}(A) + gain_{T_{i-1}}(B)}{loss(A) + loss(B)} \le \max\left\{\frac{gain_{T_{i-1}}(A)}{loss(A)}, \frac{gain_{T_{i-1}}(B)}{loss(B)}\right\}. \qquad \square$$

We define the *supergain* of a graph H with respect to a Steiner tree T as

$$supergain_T(H) = gain_T(H) + loss(H).$$

By Lemma 4.2, the supergain of K_i with respect to T_{i-1} is

$$\begin{aligned} supergain_{T_{i-1}}(K_i) &= gain_{T_{i-1}}(K_i) + loss(K_i) \\ &= cost(T_{i-1}) - mst(T_{i-1} \cup K_i) + mst(T_{i-1} \cup K_i) - cost(T_i) \\ &= cost(T_{i-1}) - cost(T_i). \end{aligned} \tag{4.2}$$

Let $G_i = supergain_{T_i}(Opt_k)$ be the supergain of the optimal k-restricted Steiner tree Opt_k with respect to T_i, $i = 0, 1, \ldots, last$. Let $loss(n)$ be the loss of the first n

accepted full trees $K_1, \ldots, K_n$. We now show that the loss of the full components identified by the k-LCA does not grow too fast.

LEMMA 4.3. *If G_n is positive, then $\frac{loss(n)}{loss_k} \le \ln \frac{G_0}{G_n}$.*

Proof. Let $l_i = loss(K_i)$ and $g_i = supergain_{T_{i-1}}(K_i)$ be, respectively, the loss and supergain of the ith full Steiner tree accepted by the k-LCA. Let Opt_k consist of full components X_j. By Lemma 2.1,

$$\frac{G_0}{loss_k} \le \frac{\sum_{X_j \in Opt_k} supergain_{T_0}(X_j)}{\sum_{X_j \in Opt_k} loss(X_j)} \le 1 + \max_{X_j \in Opt_k} \left\{ \frac{gain_{T_0}(X_j)}{loss(X_j)} \right\}$$
$$\le 1 + \frac{gain_{T_0}(K_1)}{loss(K_1)} = \frac{g_1}{l_1}.$$

Inductively, for $i = 1, 2, \ldots, n$, $\frac{G_{i-1}}{loss_k} \le \frac{g_i}{l_i}$. Therefore,

$$g_i \ge \frac{l_i}{loss_k} G_{i-1}. \tag{4.3}$$

Each time the k-LCA accepts a full tree K_i, it decreases the cost of T_i by the supergain of K_i, which results in a decrease of the supergain of Opt_k by the same value. Equality (4.2) yields $G_i = cost(T_i) - cost(Opt_k) + loss_k$. Therefore, $G_{i-1} - G_i = cost(T_{i-1}) - cost(T_i) = g_i$.

Inequality (4.3) implies that $G_i = G_{i-1} - g_i \le G_{i-1}\big(1 - \frac{l_i}{loss_k}\big)$. Since $G_n > 0$, unraveling the last inequality yields

$$\frac{G_n}{G_0} \le \prod_{i=1}^{n} \left(1 - \frac{l_i}{loss_k}\right).$$

Taking the natural logarithms of both sides and using the inequality $x \ge \ln(1+x)$, we finally obtain

$$\ln \frac{G_0}{G_n} \ge \sum_{i=1}^{n} \frac{l_i}{loss_k} = \frac{loss(n)}{loss_k}. \qquad \square \tag{4.4}$$

By Lemma 2.3, after the algorithm stops iterating, the cost of the last tree T_{last} will be at most opt_k. We stop iterating when $cost(T_{n+1}) < opt_k \le cost(T_n)$ for some n.

We now show how iteration $n+1$ can be "partially" performed so that $cost(T_{n+1})$ will *coincide* with opt_k. We split $g_{n+1} = supergain_{T_n}(K_{n+1})$ into two values g_{n+1}^1 and g_{n+1}^2 (i.e., $g_{n+1} = g_{n+1}^1 + g_{n+1}^2$) such that $cost(T_n) - g_{n+1}^1 = opt_k$ and, therefore,

$$g_{n+1}^1 = cost(T_n) - opt_k, \tag{4.5}$$

$$G_n - g_{n+1}^1 = cost(T_n) - opt_k + loss_k - (cost(T_n) - opt_k) = loss_k. \tag{4.6}$$

We split $l_{n+1} = loss(K_{n+1})$ into l_{n+1}^1 and l_{n+1}^2 such that $\frac{g_{n+1}}{l_{n+1}} = \frac{g_{n+1}^1}{l_{n+1}^1}$. Finally, we set $loss^1(n+1) = loss(n) + l_{n+1}^1$ and

$$G_{n+1}^1 = G_n - g_{n+1}^1 > 0. \tag{4.7}$$

Since $\frac{g_{n+1}}{l_{n+1}} = \frac{g^1_{n+1}}{l^1_{n+1}}$, inequality (4.4) implies that

$$\ln \frac{G_0}{G^1_{n+1}} \geq \frac{loss^1(n+1)}{loss_k}. \tag{4.8}$$

Since $g_i = gain_{T_i}(K_i) + loss(K_i) \geq loss(K_i) = l_i$, we have $\frac{g^2_{n+1}}{l^2_{n+1}} = \frac{g_{n+1}}{l_{n+1}} \geq 1$, and thus obtain

$$g^2_{n+1} \geq l^2_{n+1}. \tag{4.9}$$

The cost of the approximate Steiner tree after $n+1$ iterations is at most

$$\begin{aligned} Approx(n+1) &= mst(T_0 \cup K_1 \cup \cdots \cup K_{n+1}) \\ &\leq mst(MST(T_0 \cup K_1) \cup K_2 \cup \cdots \cup K_{n+1}) + loss(K_1) \\ &\leq mst(T_1 \cup K_2 \cup \cdots \cup K_{n+1}) + loss(K_1) \\ &\cdots \\ &\leq mst(T_n \cup K_{n+1}) + \sum_{i=1}^{n} loss(K_i) \\ &\leq cost(T_{n+1}) + loss(n+1). \qquad (4.10) \end{aligned}$$

Since $Approx(n)$ decreases with n, the upper bound on $Approx(n+1)$ also bounds $Approx = Approx(last)$, the output of the k-LCA. We complete the proof of inequality (4.1) with the following chain of inequalities:

$$\begin{aligned} Approx &\leq Approx(n+1) \\ &\leq^{(4.10)} loss(n+1) + cost(T_{n+1}) \\ &= loss(n) + l^1_{n+1} + l^2_{n+1} + cost(T_n) - g^1_{n+1} - g^2_{n+1} \\ &\leq^{(4.9)} loss(n) + l^1_{n+1} + cost(T_n) - g^1_{n+1} \\ &=^{(4.5)} loss(n) + l^1_{n+1} + opt_k \\ &\leq^{(4.8)} loss_k \cdot \ln \frac{G_0}{G^1_{n+1}} + opt_k \\ &=^{(4.7)} loss_k \cdot \ln \frac{mst - opt_k + loss_k}{G_n - g^1_{n+1}} + opt_k \\ &=^{(4.6)} loss_k \cdot \ln \frac{mst - opt_k + loss_k}{loss_k} + opt_k \\ &= loss_k \cdot \ln \left(1 + \frac{mst - opt_k}{loss_k}\right) + opt_k. \qquad \square \end{aligned}$$

5. Performance of the *k*-LCA in general graphs. Our estimate of the performance ratio of the k-LCA in arbitrary graphs is based on estimating optimal k-restricted Steiner trees. Let ρ_k be the worst-case ratio of $\frac{opt_k}{opt}$. It was shown in [6] that $\rho_k \leq 1 + \lfloor \log_2 k \rfloor^{-1}$. We will show below that the approximation ratio of the k-LCA is at most $\rho_k(1 + \frac{1}{2}\ln(\frac{4}{\rho_k} - 1))$. Therefore, the approximation ratio of the k-LCA converges to $1 + \frac{\ln 3}{2} < 1.55$ when $k \to \infty$ since $\lim_{k\to\infty} \rho_k = 1$. This is an improvement over the algorithm given by Hougardy and Prömel [10], where the approximation ratio approaches 1.59.

THEOREM 5.1. *The k-LCA has an approximation ratio of at most* $(1 + \frac{1}{2}\ln(\frac{4}{\rho_k} - 1))\rho_k$.

Proof. Since $mst \le 2 \cdot opt$ (see [21]), inequality (4.1) yields the following upper bound on the output tree cost of the k-LCA:

$$Approx \le loss_k \cdot \ln\left(1 + \frac{2 \cdot opt - opt_k}{loss_k}\right) + opt_k.$$

Following [15], we show that for any Steiner tree T, $loss(T) \le \frac{1}{2}cost(T)$. Without loss of generality, we can assume that T is a rooted tree, where all Steiner points have degree at least 3 (degree-2 Steiner points can be disregarded since the graph is complete). For each Steiner point in T, choose the shortest outgoing edge; then, all chosen edges (i) connect all Steiner points to terminals (thus having cost of at least $loss(T)$), and (ii) have total cost of at most half the cost of T. Therefore

$$loss_k \le \frac{1}{2}opt_k.$$

The partial derivative $(loss_k \cdot \ln(1 + \frac{2 \cdot opt - opt_k}{loss_k}))'_{loss_k}$ is always positive; the upper bound on *Approx* is therefore maximized when $loss_k = \frac{1}{2}opt_k$. We thus obtain

$$\frac{Approx}{opt} \le \frac{opt_k}{opt} \cdot \left(1 + \frac{\ln\left(\frac{4opt}{opt_k} - 1\right)}{2}\right).$$

Since the upper bound above grows when opt_k increases, we can replace $\frac{opt_k}{opt}$ with the maximum value of ρ_k. $\square$

6. Steiner trees in both quasi-bipartite graphs and complete graphs with edge weights 1 and 2. Recently, Rajagopalan and Vazirani [19] suggested a primal-dual–based algorithm for approximating Steiner trees. They show that their algorithm has an approximation ratio of $1.5 + \epsilon$ for quasi-bipartite graphs, i.e., the graphs in which no nonterminals are adjacent. We first show that the well-known iterated 1-Steiner heuristic [13, 9, 14] has an approximation ratio of 1.5. Next, we apply the k-LCA to quasi-bipartite graphs and estimate its runtime. Finally, we prove that the performance ratio of the k-LCA for quasi-bipartite graphs is below 1.28.

We also apply the k-LCA to the Steiner tree problem in complete graphs with edge weights 1 and 2. Bern and Plassmann [5] proved that this problem is MAX SNP-hard and gave a $\frac{4}{3} \cdot OPT$ approximation algorithm. Applying Lovász's algorithm for the parity matroid problem (see [16]), Berman, Fürer, and Zelikovsky gave a 1.2875-approximation algorithm that was given in [4]. We will show that the performance ratio of the k-LCA approaches 1.28 for such graphs, improving on previously achievable bounds.

6.1. The iterated 1-Steiner heuristic. The iterated 1-Steiner heuristic (see [13, 9, 14]) repeatedly adds Steiner points to the terminal set, as long as doing so decreases the cost of the minimum spanning tree over the terminals. Accepted Steiner points are deleted if they become useless, i.e., if their degree becomes 1 or 2 in the minimum spanning tree over the terminals. A generalization of the iterated 1-Steiner heuristic to arbitrary graphs, along with a polynomial-time implementation, is given in [1].

Although the iterated 1-Steiner heuristic decreases the minimum spanning tree cost by the maximum possible value at each iteration, we will estimate the cost of the output Steiner tree regardless of how it was obtained. The following theorem will also enable us to estimate the performance ratio of a faster *batched* variant of the iterated 1-Steiner heuristic [13, 9, 14].

THEOREM 6.1. *Given an instance of the Steiner tree problem in a quasi-bipartite graph G, let H be a Steiner tree in G such that* (i) *any Steiner point has degree at least* 3*, and* (ii) *H cannot be improved by adding any other Steiner point, i.e., $mst(H \cup v) \geq cost(H)$ for any vertex v in G. Then the cost of H is at most* 1.5 *times the optimal.*

Proof. Any full component in quasi-bipartite graphs has only a single Steiner point. Therefore, the loss of any full component equals the cost of the cheapest edge connecting its single Steiner point to a terminal. Since any Steiner point has degree at least 3 (condition (i)), the loss of any full component in H is at most one-third of its cost. Thus, $loss(H) \leq \frac{1}{3} \cdot cost(H)$.

We now show that $gain_{C[H]}(K) \leq 0$ for any full component K. Condition (ii) implies that $mst(H \cup K) \geq cost(H)$. If we contract the loss of H, then we can decrease $MST(H \cup K)$ by at most $loss(H)$ since reduction by $loss(H)$ happens only if all edges of $Loss(H)$ belong to $MST(H \cup K)$. Therefore, $mst(C[H] \cup K) \geq mst(H \cup K) - loss(H)$ and $mst(C[H] \cup K) \geq cost(H) - loss(H) = cost(C[H])$. Thus, $gain_{C[H]}(K) \leq cost(C[H]) - mst(C[H] \cup K) \leq 0$. By Lemma 2.3, $cost(H) - loss(H) \leq opt$, and since $loss(H) \leq \frac{1}{3} \cdot cost(H)$, we obtain $cost(H) \leq \frac{3}{2} \cdot opt$. □

The above result helps to explain why the iterated 1-Steiner and Rajagopalan–Vazirani heuristics perform similarly when applied to instances of the Steiner tree problem restricted to the rectilinear plane (see [17]).

6.2. Runtime of the k-LCA in quasi-bipartite graphs. For a given Steiner point v, the k-LCA adds only a full component with the largest gain, since the loss is determined by v. We can find a full tree with maximum gain with respect to a terminal-spanning tree T, among *all* possible full components with Steiner point v, by merely finding all neighbors of v in $MST(T \cup v)$. Therefore, a full component maximizing the gain-over-loss ratio over all k can be found within polynomial time.

We estimate the runtime of the k-LCA for quasi-bipartite graphs as follows. Let m and n be the number of terminals and nonterminals, respectively. The number of iterations is $O(n)$ since a Steiner point can be added only once into H. Each iteration consists of $O(n)$ gain evaluations, each of which can be computed within $O(m)$ time. Using the appropriate data structures, the k-LCA can be implemented within a total runtime of $O(n^2 \cdot m)$, where m is the number of terminals and n is the number of nonterminals.

6.3. Performance bound of the k-LCA for special graphs. We first estimate the loss of a Steiner tree in quasi-bipartite graphs and in complete graphs with edge weights 1 and 2.

LEMMA 6.2. *For the Steiner tree problem in quasi-bipartite graphs and in complete graphs with edge weights* 1 *and* 2*,*

$$mst \leq 2(opt_k - loss_k). \tag{6.1}$$

Proof. For quasi-bipartite graphs, let K be an arbitrary full component of a Steiner tree T with p terminals connected by a single Steiner point with edges of lengths $d_0, d_1, \ldots, d_{p-1}$. Assume that $loss(K) = d_0 = \min\{d_i\}$. Let $mst(K)$ be the cost of a minimum spanning tree of $G_{S'}$, where S' is the set of terminals in K. By the triangle inequality, we have

$$mst(K) \leq \sum_{i=1}^{p-1}(d_0 + d_i) = p \cdot d_0 + cost(K) - 2d_0 \leq 2cost(K) - 2loss(K).$$

The bound (6.1) follows from the fact that mst, the cost of a minimum spanning tree over S, does not exceed the sum of mst-costs for terminals in each of the full components in Opt_k.

Now we prove the lemma for the case of complete graphs with edge weights 1 and 2. Let m and n, respectively, be the number of terminals and Steiner points in the optimal k-restricted Steiner tree Opt_k. Then $mst \leq 2m - 2$ since all edge weights are at most 2, and $opt_k \geq m + n - 1$ since Opt_k contains $m + n$ nodes. We may assume that full components of Opt_k contain only edges of weight 1, and therefore $loss_k = n$. Thus, $mst \leq 2m - 2 = 2(m + n - 1 - n) \leq 2(opt_k - loss_k)$. □

THEOREM 6.3. *The k-LCA has an approximation ratio of at most* ≈ 1.279 *for quasi-bipartite graphs and an approximation ratio approaching* ≈ 1.279 *for complete graphs with edge weights* 1 *and* 2.

Proof. After substituting the minimum spanning tree bound (6.1) into inequality (4.1), we obtain

$$Approx \leq loss_k \cdot \ln\left(\frac{opt_k}{loss_k} - 1\right) + opt_k. \tag{6.2}$$

Taking the partial derivative of $(loss \cdot \ln(\frac{opt_k}{loss_k} - 1))'_{loss_k}$, we see that the single maximum of the upper bound (6.2) occurs when $x = \frac{loss_k}{opt_k - loss_k}$ is the root of the equation $1 + \ln x + x = 0$. Solving this equation numerically, we obtain $x \approx 0.279$. Finally, we substitute x into (6.2), yielding

$$Approx \leq \frac{x}{1+x} \cdot opt_k \cdot \ln\frac{1}{x} + opt_k = (x+1) \cdot opt_k \approx 1.279 \cdot opt_k.$$

The bound above is valid for the output of the k-LCA for quasi-bipartite graphs if we set $k = |S|$, i.e., if we omit the index k. For complete graphs with edge weights 1 and 2, opt_k converges to opt, and the approximation ratio of the k-LCA therefore converges to 1.279 when $k \to \infty$. □

7. Conclusions and open problems. We presented a new best-performing polynomial-time heuristic for the classical graph Steiner tree problem. This heuristic, called the k-restricted loss-contracting algorithm (k-LCA), can be applied to arbitrary metric spaces. The worst-case performance for the k-LCA depends on the Steiner ratio and the loss of the optimal Steiner tree (i.e., the cost of connecting Steiner points to terminals). We proved that the k-LCA is currently the best approximation heuristic for the Steiner tree problem in graphs: its approximation ratio is ≈ 1.55 for general graphs and ≈ 1.28 for both quasi-bipartite graphs and graphs with edge costs 1 and 2. We also used our techniques to derive the first known nontrivial performance ratio ($1.5 \cdot$ OPT) for the iterated 1-Steiner heuristic of Kahng and Robins [13, 9, 14, 1] in quasi-bipartite graphs.

Chief among the remaining open problems is finding heuristics for the classical graph Steiner problem with improved performance bounds. Other special cases of the Steiner problem for special metrics, cost functions, and graph types may be explored separately, where it may be possible to exploit the specific underlying structure to further improve the performance bounds. Interestingly, our k-LCA is the first (and so far the only) heuristic that is proven to work well for all of the special graph types discussed above.

From a practical perspective, for any given fixed performance bound it would be useful to minimize the running times of the associated heuristics and to quantify and explore various tradeoffs between running times and solution quality. Finally, it would be useful to implement the various heuristics and explore their practical runtime and empirical solution quality by comparing the performance of these implementations side by side on various classes and sizes of inputs.

Acknowledgments. We thank Gruia Călinescu for reading earlier drafts of this paper and giving numerous helpful suggestions.

REFERENCES

[1] M. J. Alexander and G. Robins, *New performance-driven FPGA routing algorithms*, IEEE Trans. Comput. Aided Design Integrated Circuits and Systems, 15 (1996), pp. 1505–1517.

[2] S. Arora, *Polynomial time approximation schemes for Euclidean TSP and other geometric problems*, J. ACM, 45 (1998), pp. 753–782.

[3] P. Berman and V. Ramaiyer, *Improved approximations for the Steiner tree problem*, J. Algorithms, 17 (1994), pp. 381–408.

[4] P. Berman, M. Fürer, and A. Zelikovsky, *Applications of the Matroid Parity Problem to Approximating Steiner Trees*, Tech. report 980021, Computer Science Department, UCLA, Los Angeles, 1998. Available online at ftp://ftp.cs.ucla.edu/tech-report/1998-reports.

[5] M. Bern and P. Plassmann, *The Steiner tree problem with edge lengths* 1 *and* 2, Inform. Process. Lett., 32 (1989), pp. 171–176.

[6] A. Borchers and D.-Z. Du, *The k-Steiner ratio in graphs*, SIAM J. Comput., 26 (1997), pp. 857–869.

[7] A. Caldwell, A. B. Kahng, S. Mantik, I. Markov, and A. Zelikovsky, *On wirelength estimations for row-based placement*, in Proceedings of the International Symposium on Physical Design, ACM, New York, 1998, pp. 4–11.

[8] M. R. Garey and D. S. Johnson, *The rectilinear Steiner tree problem is NP-complete*, SIAM J. Appl. Math., 32 (1977), pp. 826–834.

[9] J. Griffith, G. Robins, J. S. Salowe, and T. Zhang, *Closing the gap: Near-optimal Steiner trees in polynomial time*, IEEE Trans. Comput. Aided Design Integrated Circuits and Systems, 13 (1994), pp. 1351–1365.

[10] S. Hougardy, and H. J. Prömel, *A* 1.598 *approximation algorithm for the Steiner problem in graphs*, in Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, ACM, New York, 1999, pp. 448–453.

[11] F. K. Hwang, D. S. Richards, and P. Winter, *The Steiner Tree Problem*, North-Holland, Amsterdam, 1992.

[12] B. Korte, H. J. Prömel, and A. Steger, *Steiner trees in VLSI-layout*, in Paths, Flows, and VLSI-Layout, B. Korte et al., eds., Springer, Berlin, 1990, pp. 185–214.

[13] A. B. Kahng and G. Robins, *A new class of iterative Steiner tree heuristics with good performance*, IEEE Trans. Comput. Aided Design Integrated Circuits and Systems, 11 (1992), pp. 893–902.

[14] A. B. Kahng and G. Robins, *On Optimal Interconnections for VLSI*, Kluwer, Dordrecht, 1995.

[15] M. Karpinski and A. Zelikovsky, *New approximation algorithms for the Steiner tree problem*, J. Combin. Optim., 1 (1997), pp. 47–65.

[16] L. Lovász and M. D. Plummer, *Matching Theory*, Elsevier Science, Amsterdam, 1986.

[17] I. I. Mandoiu, V. V. Vazirani, and J. L. Ganley, *A new heuristic for rectilinear Steiner trees*, IEEE Trans. Comput. Aided Design Integrated Circuits and Systems, 19 (2000), pp. 1129–1139.

[18] H. J. Prömel and A. Steger, *RNC-approximation algorithms for the Steiner problem*, in Proceedings of the 14th Annual Symposium on Theoretical Aspects of Computer Science, Springer, Berlin, 1997, pp. 559–570.

[19] S. Rajagopalan and V. V. Vazirani, *On the bidirected cut relaxation for the metric Steiner tree problem*, in Proceedings of Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, ACM, New York, 1999, pp. 742–751.

[20] G. Robins and A. Zelikovsky, *Improved Steiner tree approximation in graphs*, in Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, ACM, New York, 2000, pp. 770–779.

[21] H. Takahashi and A. Matsuyama, *An approximate solution for the Steiner problem in graphs*, Math. Jap., 24 (1980), pp. 573–577.

[22] A. Zelikovsky, *An* 11/6*-approximation algorithm for the network Steiner problem*, Algorithmica, 9 (1993), pp. 463–470.

[23] A. Zelikovsky, *Better Approximation Bounds for the Network and Euclidean Steiner Tree Problems*, Tech. report CS-96-06, University of Virginia, Charlottesville, VA, 1996.

SIAM J. Discrete Math.
Vol. 19, No. 1, pp. 135–144

HAMILTONIAN CYCLES WITH PRESCRIBED EDGES IN HYPERCUBES[*]

TOMÁŠ DVOŘÁK[†]

Abstract. Given a set $\mathcal{P}$ of at most $2n-3$ prescribed edges ($n \geq 2$), the n-dimensional hypercube Q_n contains a Hamiltonian cycle passing through all edges of $\mathcal{P}$ iff the subgraph induced by $\mathcal{P}$ consists of pairwise vertex-disjoint paths. This answers a question of Caha and Koubek, who showed that for any $n \geq 3$ there are $2n-2$ edges of Q_n not contained in any Hamiltonian cycle, but that still satisfy the above condition.

Key words. Hamiltonian cycles passing through given edges, hypercube

AMS subject classifications. 05C38, 05C45, 68M10, 68M15, 68R10

DOI. 10.1137/S0895480103432805

1. Introduction.

The n-dimensional hypercube Q_n is a graph whose vertex set consists of all binary vectors of length n, with two vertices being adjacent whenever the corresponding vectors differ in exactly one coordinate. There is a large amount of literature on graph-theoretic properties of hypercubes (e.g., see the comprehensive survey paper [7]) as well as on their applications in parallel computing (e.g., see [10], a monograph partially devoted to hypercubic architectures of massively parallel computers).

It is well known that Q_n is Hamiltonian for every $n \geq 2$. The publication of this fact dates back to 1872 [5], and since then the research on Hamiltonian cycles in hypercubes satisfying certain additional properties has received considerable attention (a survey can be found in [11]). The applications in parallel computing inspired the study of hypercubes with faulty links, which lead to the investigation of Hamiltonian cycles of Q_n avoiding a certain set of forbidden edges [2, 3, 12].

This paper studies the following problem, which is in a sense complementary to the last one mentioned above: Given a set of prescribed edges in the hypercube, which conditions guarantee the existence of a Hamiltonian cycle passing through every edge of this set? This question has been proposed in a recent work by Caha and Koubek [1], where they observed that *any proper subset $\mathcal{P}$ of edges of a Hamiltonian cycle necessarily induces a subgraph consisting of pairwise vertex-disjoint paths.* They showed that in the case when $|\mathcal{P}| \leq n-1$, $n \geq 2$, this condition is also sufficient to guarantee the existence of a Hamiltonian cycle of Q_n passing through every edge of $\mathcal{P}$. On the other hand, for any $n \geq 3$ there is a set of $2n-2$ edges, satisfying the above condition, but not contained in any Hamiltonian cycle. Indeed, let v be an arbitrary vertex of Q_n and let $\mathcal{P}$ be a set of edges incident with neighbors of v so that all neighbors, except one neighbor of v, are incident with exactly two edges of $\mathcal{P}$, but no edge of $\mathcal{P}$ is incident with v. It is not difficult to see that this always can be done in such a way that the above condition is preserved. Since Q_n is a regular graph of degree n, it follows that $|\mathcal{P}| = 2n-2$ and that any cycle passing through all edges of $\mathcal{P}$ avoids v.

[*] Received by the editors August 5, 2003; accepted for publication (in revised form) November 3, 2004; published electronically June 16, 2005.
http://www.siam.org/journals/sidma/19-1/43280.html

[†] Faculty of Mathematics and Physics, Charles University, Malostranské nám. 25, 118 00 Praha, Czech Republic (Tomas.Dvorak@mff.cuni.cz).

The main purpose of this paper is to show that the upper bound obtained in this way is sharp: Given a set of at most $2n-3$ prescribed edges of Q_n, $n \geq 2$, satisfying the above necessary condition, we describe a construction of a Hamiltonian cycle passing through every edge of this set.

2. Terminology and notation. The terminology and notation used in this paper mostly follow [6]. As usual, the vertex and edge sets of a graph G are denoted by $V(G)$ and $E(G)$, respectively. The distance of vertices x and y in G is denoted by $d_G(x,y)$, with the subscript being omitted when the context is clear. For $e, e' \in E(G)$, $d_G(e,e') = \min\{d_G(x,y) \mid x \in e, y \in e'\}$. Given a set $\mathcal{E} \subseteq E(G)$ of edges of a graph G, $\langle \mathcal{E} \rangle$ denotes the subgraph of G, induced by $\mathcal{E}$, i.e., $V(\langle \mathcal{E} \rangle) = \bigcup_{e \in \mathcal{E}} e$, $E(\langle \mathcal{E} \rangle) = \mathcal{E}$.

A *path* $P = x_0, x_1, \ldots, x_n$ of length n between x_0 and x_n is a graph P with $V(P) = \{x_0, \ldots, x_n\}$ and $E(P) = \{\{x_0,x_1\}, \{x_1,x_2\}, \ldots, \{x_{n-1},x_n\}\}$. Vertices x_0 and x_n are called *endvertices* of P. A *cycle* of length n is a connected 2-regular graph on n vertices. Given a set $\mathcal{E}$ of edges, a path P (cycle C) *passes through* $\mathcal{E}$ if $\mathcal{E} \subseteq E(P)$ ($\mathcal{E} \subseteq E(C)$). The vertices of $V(P)$ ($V(C)$) form the vertex set *spanned* by path P (cycle C).

The following properties and notation shall be useful later:

1. For any $n \geq 1$ there are $n+1$ ways to split Q_{n+1} into two disjoint copies of Q_n, denoted by Q_n^L and Q_n^R (the "left" and the "right" subcube). Then any vertex $x \in V(Q_n^L)$ has in Q_n^R a unique neighbor, denoted by x^R. Similarly, any vertex $y \in V(Q_n^R)$ has in Q_n^L a unique neighbor, denoted by y^L. For an edge $e = \{x,y\} \in E(Q_n^L)$, e^R denotes the edge $\{x^R, y^R\} \in E(Q_n^R)$.

Given a set $\mathcal{P} \subseteq E(Q_{n+1})$, we denote its subsets $\mathcal{P} \cap E(Q_n^L)$ and $\mathcal{P} \cap E(Q_n^R)$ by $\mathcal{P}^L$ and $\mathcal{P}^R$, respectively.

2. If x, y, z are pairwise distinct vertices of Q_{n+1} such that x and y are adjacent, there exists a split such that $x, y \in V(Q_n^L)$ and $z \in V(Q_n^R)$.

3. Q_n is a bipartite graph for any $n \geq 1$, which means that for any $x, y \in V(Q_n)$, the length of each path between x and y has the same parity as $d(x,y)$.

4. The (0,2)-property: Any two distinct vertices of Q_n ($n \geq 1$) have exactly two neighbors in common or none at all.

5. For any edge $e \in E(Q_3)$ there exists a unique edge $e' \in E(Q_3)$ such that $d(e,e') = 2$.

6. The notation $A = B \dot{\cup} C$ means that $A = B \cup C$ and $B \cap C = \emptyset$.

3. Lemmas. This section is devoted to auxiliary results preparing the necessary technique for the constructive proof of the main theorem, which is based on the usual divide and conquer strategy: Split the hypercube into two subcubes, inductively construct cycles in each part, and join them to obtain the desired Hamiltonian cycle of the whole graph. The following lemma guarantees that the split always can be done in such a way that it enables the cycles to be combined in the final one.

Given a set of prescribed edges $\mathcal{P}$ and a split of Q_{n+1} into Q_n^L and Q_n^R, we say that an edge $e \in E(Q_n^L)$ is *free* if $e \notin \mathcal{P}^L$, $e^R \notin \mathcal{P}^R$, and both $\langle \mathcal{P}^L \cup \{e\} \rangle$ and $\langle \mathcal{P}^R \cup \{e^R\} \rangle$ consist of pairwise vertex-disjoint paths.

LEMMA 3.1. *Let $n \geq 3$ and $\mathcal{P} \subseteq E(Q_{n+1})$ such that $|\mathcal{P}| \leq 2(n+1)-3$ and $\langle \mathcal{P} \rangle$ consists of pairwise vertex-disjoint paths. Then there exists a split of Q_{n+1} into subcubes Q_n^L and Q_n^R satisfying the following conditions:*

(i) *There exists a free edge $\{x,y\}$ in Q_n^L;*

(ii) *$\mathcal{P} \setminus (\mathcal{P}^L \cup \mathcal{P}^R)$ is either empty or equals $\{\{x, x^R\}\}$.*

Proof. First, assume that Q_{n+1} is split into subcubes Q_n^L and Q_n^R such that $\mathcal{P} = \mathcal{P}^L \cup \mathcal{P}^R$. Observe that $e \in E(Q_n^L)$ is not free iff e or e^R

- belongs to $\mathcal{P}$, or
- is incident with a vertex of $\langle \mathcal{P} \rangle$ of degree two, or
- joins endvertices of a path of $\langle \mathcal{P} \rangle$.

Denote by m the number of vertices of $\langle \mathcal{P} \rangle$ of degree two and by p the number of connected components of $\langle \mathcal{P} \rangle$ and observe that $m \leq |\mathcal{P}| - 1$ and $m + p = |\mathcal{P}|$. Consequently, $|\{e \in E(Q_n^L) \mid e \text{ is not free}\}|$ does not exceed

$$|\mathcal{P}| + m(n-2) + p = 2|\mathcal{P}| + m(n-3) \leq 2(n-1)^2 + 2 < n2^{n-1} = |E(Q_n)|$$

for $n \geq 3$. Hence there must be a free edge in Q_n^L.

Next assume that $\mathcal{P} \setminus (\mathcal{P}^L \cup \mathcal{P}^R) = \{\{x, x^R\}\}$ for some $x \in V(Q_n^L)$. If condition (i) does not hold, then for any edge $\{x, y\} \in E(Q_n^L)$, at least one of the following must be true:

(a) y is an endvertex of a path, starting at x, which forms a connected component of $\langle \mathcal{P}^L \rangle$;

(b) y is incident with two edges of $\mathcal{P}^L$;

(c) y^R is an endvertex of a path, starting at x^R, which forms a connected component of $\langle \mathcal{P}^R \rangle$;

(d) y^R is incident with two edges of $\mathcal{P}^R$.

Note that as any hypercube is a triangle-free graph, no edge of $\mathcal{P}^L$ ($\mathcal{P}^R$) can be incident with two distinct neighbors of x (x^R). Moreover, since x has exactly n distinct neighbors in Q_n^L and each of conditions (a) and (c) holds for at most one of them, it follows that there are at least $2 + 2(n-2) = 2n - 2$ edges of $\mathcal{P}^L \cup \mathcal{P}^R$, incident either with a neighbor of x in Q_n^L or with a neighbor of x^R in Q_n^R. Taking into account that $\{x, x^R\} \in \mathcal{P}$, it follows that $|\mathcal{P}| \geq 2n - 1$. Since we assumed that $|\mathcal{P}| \leq 2n - 1$, we can conclude that this inequality is actually equality. Hence

(e) every edge of $\mathcal{P}^L$ ($\mathcal{P}^R$) must be incident with a neighbor of x (x^R).

Moreover,

(f) for any neighbor y of x in Q_n^L, exactly one of conditions (a)–(d) holds.

On the other hand,

(g) each of conditions (a) and (c) holds for exactly one neighbor of x.

In particular, it follows (from (g)) that there are paths $u_0 = x, u_1, \ldots, u_{2k+1}$ and $v_0 = x^R, v_1, \ldots, v_{2r+1}$ forming connected components of $\langle \mathcal{P}^L \rangle$ and $\langle \mathcal{P}^R \rangle$, respectively, such that (by (e)) for any $i \in \{0, 1, \ldots, k\}$ and $j \in \{0, 1, \ldots, r\}$, u_{2i+1} is a neighbor of x, v_{2j+1} is a neighbor of x^R, and $d(u_{2i+1}, v_{2j+1}) = 3$ (there is a path of length three between them, and the distance cannot be one, since then $v_{2j+1} = u_{2i+1}^R$, contrary to (f)). As $\{x, x^R\} \in \mathcal{P}$, path $P = u_{2k+1}, u_{2k}, \ldots, u_1, u_0, v_0, v_1, \ldots, v_{2r}, v_{2r+1}$ forms a connected component of $\langle \mathcal{P} \rangle$. We can conclude that

($*$) one of the connected components of $\langle \mathcal{P} \rangle$ is a path P of odd length such that the only edge of P, incident with neighbors of both endvertices of P, is $\{x, x^R\}$.

What can be said about an arbitrary path P', $P' \neq P$, which forms a connected component of $\langle \mathcal{P} \rangle$? First note that $\{x, x^R\} \notin E(P')$, as it already belongs to P. Hence P' is either in $\langle \mathcal{P}^L \rangle$ or in $\langle \mathcal{P}^R \rangle$. Now choose an arbitrary edge e of P' and observe that by condition (e) there has to be a neighbor y of x in Q_n^L such that e is incident with y or y^R. By (f), exactly one of conditions (a)–(d) holds for y. Condition (g) reveals that it cannot be (a) and (c), as they already hold for endvertices of P, which is vertex-disjoint with P'. It follows that y has to be incident with exactly two edges of P'. Hence P' is a path of even length. It follows that

$(**)$ P is the only path of odd length which forms a connected component of $\langle\mathcal{P}\rangle$; moreover, $\{x, x^R\} \in E(P)$.

Now assume that none of the $n+1$ splits of Q_{n+1} into two n-dimensional subcubes has the property that each edge of $\mathcal{P}$ belongs to one of the two subcubes forming the split. If there were at least n distinct splits of Q_{n+1} into Q_n^L and Q_n^R such that for each of them, $|\mathcal{P} \setminus (\mathcal{P}^L \cup \mathcal{P}^R)| \geq 2$, then

$$|\mathcal{P}| \geq 2n + 1 > 2(n+1) - 3,$$

contrary to our assumption. It follows that there must exist two distinct splits into subcubes $Q_n^{L_i}$ and $Q_n^{R_i}$ ($i \in \{1,2\}$) such that $\mathcal{P} \setminus (\mathcal{P}^{L_i} \cup \mathcal{P}^{R_i}) = \{\{x_i, x_i^R\}\}$. Suppose, by way of contradiction, that none of them satisfies (i). Then both $(*)$ and $(**)$ must hold for each $\{x_i, x_i^R\}$, $i \in \{1,2\}$. In particular, condition $(**)$ says that there is a unique path P of odd length which forms a connected component of $\mathcal{P}$; moreover, $\{x_i, x_i^R\} \in E(P)$ for each $i \in \{1,2\}$. Furthermore, condition $(*)$ reveals that the only edge of P, incident with neighbors of both endvertices of P, is $\{x_i, x_i^R\}$ for each $i \in \{1,2\}$. It follows that $\{x_1, x_1^R\} = \{x_2, x_2^R\}$, which contradicts our assumption that $\{Q_n^{L_1}, Q_n^{R_1}\} \neq \{Q_n^{L_2}, Q_n^{R_2}\}$. ☐

The main obstacle we face in our divide and conquer approach is an unequal distribution of prescribed edges between two subcubes. If one of the subcubes into which the hypercube is split contains too many prescribed edges, the induction cannot be used. We propose a solution based on a simple idea: Temporarily remove some edges from the prescribed set, apply the induction, and then repair the cycle obtained in this way to include also the edges temporarily removed. In the process of repairing, the cycle may fall apart into several paths. In order to join them into the desired Hamiltonian cycle of the whole graph, we need to cover the other subcube with a certain number of paths. This task is accomplished by the following lemmas.

First let us recall the following classical result, originally proved by Havel in [8].

Lemma 3.2 (Havel's lemma). *Let $n \geq 1$ and $x, y \in V(Q_n)$ be such that $d(x,y)$ is odd. Then there exists a Hamiltonian path between x and y in Q_n.*

The following generalization is needed.

Lemma 3.3. *Let x, y, u, v be pairwise distinct vertices of Q_n, $n \geq 2$, such that both $d(x,y)$ and $d(u,v)$ are odd. Then*

(i) *there exist paths P_1 between x and y and P_2 between u and v such that $V(Q_n) = V(P_1) \dot\cup V(P_2)$;*

(ii) *moreover, in the case when $d(x,y) = 1$, path P_1 can be chosen such that $P_1 = x, y$, unless $n = 3$, $d(u,v) = 1$, and $d(\{x,y\},\{u,v\}) = 2$.*

Proof. We argue by induction on n. The case $n = 2$ is left to the kind reader, while the case $n = 3$ is settled by Figure 3.1. Now let $n \geq 3$ and assume that the statement of the theorem holds for the hypercube of dimension n. Split Q_{n+1} into Q_n^L and Q_n^R such that $x \in V(Q_n^L)$ and $v \in V(Q_n^R)$; moreover, in the case when $d(x,y) = 1$, without loss of generality, assume that y belongs to $V(Q_n^L)$ as well.

Case 1. $y \in V(Q_n^L)$.

(1.1) $u \in V(Q_n^L)$. Choose a vertex $w \in V(Q_n^L) \setminus \{x, y, u\}$ such that $d(u,w)$ is odd and, in the case when $n = 3$, $d(u,w) \neq 1$. By the induction hypothesis, there are paths P_1 between x and y and P_2' between u and w whose vertices form a partition of $V(Q_n^L)$. Moreover, in the case when $d(x,y) = 1$ we can assume that $P_1 = x, y$. Since $d(u, w^R)$ is even, $d(w^R, v)$ must have the same parity as $d(u,v)$, which is odd by our assumption. Hence by Lemma 3.2, there exists a Hamiltonian path P_2'' between w^R and v in Q_n^R. The desired path P_2 is a concatenation of paths P_2' and P_2''.

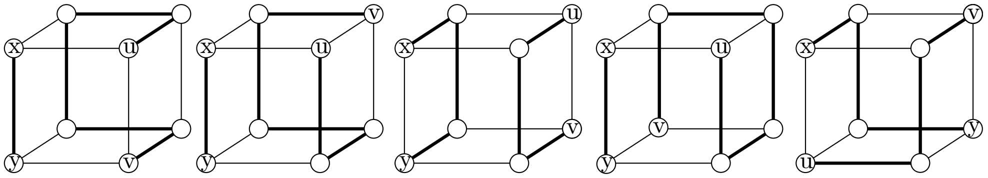


FIG. 3.1. *Case* $n = 3$ *of Lemma* 3.3.

(1.2) $u \in V(Q_n^R)$. First apply Lemma 3.2 to obtain a Hamiltonian path P_2' between u and v in Q_n^R. In the case when $d(x, y) > 1$ simply apply Lemma 3.2 again to obtain a Hamiltonian path P_1 between x and y in Q_n^L and set $P_2 = P_2'$. If, however, $d(x, y) = 1$, observe that it is possible to choose an edge $\{r, s\} \in E(P_2')$ such that $\{r^L, s^L\} \cap \{x, y\} = \emptyset$. Moreover, in the case when $n = 3$, this choice can be made in such a way that $d(\{x, y\}, \{r^L, s^L\}) \neq 2$. It remains to apply the induction to obtain paths $P_1 = x, y$ and P_2'' between r^L and s^L whose vertices partition $V(Q_n^L)$. The desired path P_2 is obtained by replacing r, s in P_2' with path r, P_2'', s.

Case 2. $y \in V(Q_n^R)$. Note that in this case $d(x, y) \geq 3$.

(2.1) $u \in V(Q_n^L)$. Choose in Q_n^L distinct vertices w and z such that both $d(u, w)$ and $d(x, z)$ are odd and $\{w, z\} \cap \{x, u\} = \emptyset = \{w^R, z^R\} \cap \{y, v\}$. By induction, there are paths P_1' between x and z and P_2' between u and w whose vertices partition $V(Q_n^L)$. Similarly, there are paths P_1'' between z^R and y and P_2'' between w^R and v whose vertices partition $V(Q_n^R)$. The desired paths P_1 and P_2 are obtained as concatenations of P_1' with P_1'' and of P_2' with P_2'', respectively.

(2.2) $u \in V(Q_n^R)$. This case is isomorphic to (1.1). □

COROLLARY 3.4. *Let* $n \geq 2$, $x, y \in V(Q_n)$, *and* $e \in E(Q_n)$ *be such that* $d(x, y)$ *is odd and* $e \neq \{x, y\}$. *Then there is a Hamiltonian path in* Q_n *between* x *and* y *passing through edge* e.

Proof. First consider the case when e is incident with one of the vertices, say $e = \{x, u\}$, $u \neq y$. Let v be an arbitrary neighbor of u, distinct from x. Since $d(x, y)$ is odd by our assumption and $d(x, v) = 2$, $d(y, v)$ must be odd as well. Note that this also means that vertices x, y, u, and v are pairwise disjoint. Moreover, in the case when $n = 3$ and $d(y, v) = 1$, the fact that $d(u, v) = 1$ implies $d(\{x, u\}, \{y, v\}) \neq 2$. Hence, it is safe to apply part (ii) of Lemma 3.3 to obtain paths $P_1 = x, u$ and P_2 between v and y, whose concatenation provides the desired Hamiltonian path between x and y passing through e.

It remains to settle the case when $e = \{u, v\}$ such that $\{u, v\} \cap \{x, y\} = \emptyset$. Observe that then necessarily either both $d(x, u)$ and $d(y, v)$ or both $d(x, v)$ and $d(y, u)$ must be odd. The statement of the corollary now follows from part (i) of Lemma 3.3. □

It should be noted that Corollary 3.4 also follows from a more general theorem of [1] on paths with prescribed edges in hypercubes.

The last lemma of this section gathers the fruit of our previous work by showing how to extend paths in one subcube to a Hamiltonian cycle of the whole graph.

LEMMA 3.5. *Let* $n \geq 1$, Q_{n+1} *be split into subcubes* Q_n^L *and* Q_n^R, *and* $\mathcal{P} \subseteq E(Q_n^L)$ *be such that* $\langle \mathcal{P} \rangle$ *either forms a Hamiltonian path of* Q_n^L *or consists of paths* P_1 *between* x *and* y *and* P_2 *between* u *and* v *in* Q_n^L *such that*

(i) $V(Q_n^L) = V(P_1) \,\dot\cup\, V(P_2)$;

(ii) *there are two vertices in* $\{x, y, u, v\}$ *whose mutual distance is odd.*

Then there exists a Hamiltonian cycle of Q_{n+1} *passing through* $\mathcal{P}$.

Proof. If $\langle \mathcal{P} \rangle$ forms a Hamiltonian path between x and y in Q_n^L, find an isomorphic path between x^R and y^R in Q_n^R and join both paths by edges $\{x, x^R\}$ and $\{y, y^R\}$.

Next assume that there are paths P_1 and P_2 such that conditions (i) and (ii) hold. First observe that as $|V(Q_n)|$ is even, $|V(P_1)| \equiv |V(P_2)| \pmod 2$, and hence $d(x, y)$ and $d(u, v)$ have the same parity. In our connected bipartite graph it means that $d(x, u)$ and $d(y, v)$ have the same parity, too. It is easy to see that condition (ii) and bipartiteness imply that both $d(x, u)$ and $d(y, v)$ are odd, or both $d(x, v)$ and $d(y, u)$ are odd. Without a loss of generality, we can assume the former.

It remains to use Lemma 3.3 to construct paths P_1' between x^R and u^R and P_2' between y^R and v^R in Q_n^R so that $V(Q_n^R) = V(P_1') \,\dot\cup\, V(P_2')$ and to observe that the desired Hamiltonian cycle of Q_{n+1} is induced by edges of $E(P_1) \cup E(P_2) \cup E(P_1') \cup E(P_2') \cup \{\{x, x^R\}, \{y, y^R\}, \{u, u^R\}, \{v, v^R\}\}$. □

4. Hamiltonian cycles. Now we are ready to prove the main result.

THEOREM 4.1. *Let $n \geq 2$ and $\mathcal{P} \subseteq E(Q_n)$ be such that $|\mathcal{P}| \leq 2n - 3$. Then there exists a Hamiltonian cycle of Q_n passing through $\mathcal{P}$ iff $\langle \mathcal{P} \rangle$ consists of pairwise vertex-disjoint paths.*

Proof. The necessity of the condition (observed in [1]) follows from the fact that for any Hamiltonian cycle C of Q_n, $|\mathcal{P}| < |V(Q_n)| = |E(C)|$ and a proper subset of edges of a cycle always induces a graph consisting of vertex-disjoint paths. It remains to show that the condition is also sufficient. We argue by induction on n. The case $n \in \{2, 3\}$ may be verified by inspection (which is done in detail in [1]). Now let $n \geq 3$ and assume that the statement of the theorem holds for the hypercube of dimension n. Using Lemma 3.1, split Q_{n+1} into subcubes Q_n^L and Q_n^R such that $|\mathcal{P} \setminus (\mathcal{P}^L \cup \mathcal{P}^R)| \leq 1$. Assuming without a loss of generality that $|\mathcal{P}^L| \geq |\mathcal{P}^R|$, consider the following two cases.

Case 1. There exists $x \in V(Q_n^L)$ such that $\mathcal{P} \setminus (\mathcal{P}^L \cup \mathcal{P}^R) = \{\{x, x^R\}\}$.

(1.1) $|\mathcal{P}^L| < 2n - 3$. By Lemma 3.1 we can assume that there is a free edge $\{x, y\} \in E(Q_n^L)$. By the induction hypothesis there exist Hamiltonian cycles C in Q_n^L and C' in Q_n^R passing through $\mathcal{P}^L \cup \{x, y\}$ and $\mathcal{P}^R \cup \{x^R, y^R\}$, respectively. The desired Hamiltonian cycle of Q_{n+1} is induced by edges of $(E(C) \cup E(C') \cup \{\{x, x^R\}, \{y, y^R\}\}) \setminus \{\{x, y\}, \{x^R, y^R\}\}$.

(1.2) $|\mathcal{P}^L| = 2n - 3$ and $|\mathcal{P}^R| \leq 1$. Using the induction, find a Hamiltonian cycle C in Q_n^L passing through $\mathcal{P}^L$. Since $\{x, x^R\} \in \mathcal{P}$, at most one of the two edges of C incident with x may belong to $\mathcal{P}^L$. Hence there has to be an edge $\{x, y\} \in E(C) \setminus \mathcal{P}^L$. If $\{x^R, y^R\} \notin \mathcal{P}^R$, then simply use the induction hypothesis to find in Q_n^R a Hamiltonian cycle C' containing $\mathcal{P}^R \cup \{\{x^R, y^R\}\}$; the desired Hamiltonian cycle of Q_{n+1} is then induced by edges of $(E(C) \cup E(C') \cup \{\{x, x^R\}, \{y, y^R\}\}) \setminus \{\{x, y\}, \{x^R, y^R\}\}$.

If, however, $\{x^R, y^R\} \in \mathcal{P}^R$, choose an edge $\{r, s\} \in E(C) \setminus \mathcal{P}^L$ such that $\{r, s\} \cap \{x, y\} = \emptyset$. Note that as

$$|E(C) \setminus (\mathcal{P}^L \cup \{e \in E(C) \mid e \cap \{x, y\} \neq \emptyset\})| \geq |E(C)| - |\mathcal{P}^L| - 3 \geq 2^n - (2n - 3) - 3 \geq 2$$

for $n \geq 3$, there are always at least two ways to choose such an edge. In particular, in case $n = 3$ this choice can be made in such a way that $d(\{r, s\}, \{x, y\}) \neq 2$. Then $d(\{r^R, s^R\}, \{x^R, y^R\}) \neq 2$ as well, and therefore it is safe to apply part (ii) of Lemma 3.3 to find in Q_n^R a path P between r^R and s^R, spanning all vertices of Q_n^R except x^R and y^R. The desired Hamiltonian cycle of Q_{n+1} is then induced by edges of $(E(C) \cup E(P) \cup \{\{x, x^R\}, \{x^R, y^R\}, \{y, y^R\}, \{r, r^R\}, \{s, s^R\}\}) \setminus \{\{x, y\}, \{r, s\}\}$.

(1.3) $|\mathcal{P}^L| = 2n-2$ and $|\mathcal{P}^R| = 0$. First choose an edge $\{u,v\} \in \mathcal{P}^L$ such that v is an endvertex of a path of $\langle \mathcal{P}^L \rangle$; moreover, if $\langle \mathcal{P}^L \rangle$ contains a path with an endvertex equal to x, then $v = x$. Next apply the induction to find in Q_n^L a Hamiltonian cycle C passing through $\mathcal{P}^L \setminus \{\{u,v\}\}$ and consider the following two subcases.

(1.3.1) $\{u,v\} \notin C$. Let r and s be the respective neighbors of u and v on C such that

(i) $\{u,r\} \notin \mathcal{P}^L$;

(ii) each of the two paths between r and s on C contains either u or v.

The way v has been chosen implies that $\{v,s\} \notin \mathcal{P}^L$. The existence of a path of length three between r and s means that $d(r,s)$ is odd and the same has to be true about the distance of their respective neighbors r^R and s^R in Q_n^R. In case $x = r$ or $x = s$ simply use Lemma 3.2 to find a Hamiltonian path P between r^R and s^R in Q_n^R. The desired Hamiltonian cycle of Q_{n+1} is then induced by edges of $(E(C) \cup E(P) \cup \{\{r,r^R\},\{s,s^R\}\}) \setminus \{\{u,r\},\{v,s\}\}$; since $\{x,x^R\}$ equals $\{r,r^R\}$ or $\{s,s^R\}$, it really passes through $\mathcal{P}$.

If, however, x is distinct from both r and s, find on C a neighbor y of x so that $\{x,y\} \notin \mathcal{P}^L$. Note that the way $\{u,v\}$ has been defined guarantees that y can be chosen so that $y \notin \{r,s\}$. It remains to use Lemma 3.3 to partition $V(Q_n^R)$ into vertices of paths P_1 between r^R and s^R and P_2 between x^R and y^R. The desired Hamiltonian cycle of Q_{n+1} is then induced by edges of $(E(C) \cup E(P_1) \cup E(P_2) \cup \{\{u,v\},\{x,x^R\},\{y,y^R\},\{r,r^R\},\{s,s^R\}\}) \setminus \{\{x,y\},\{u,r\},\{v,s\}\}$.

(1.3.2) $\{u,v\} \in C$. Let y be a neighbor of x on C such that $\{x,y\} \notin \mathcal{P}^L$. By Lemma 3.2 Q_n^R contains a Hamiltonian path P between x^R and y^R and the desired Hamiltonian cycle of Q_{n+1} is induced by edges of $(E(C) \cup E(P) \cup \{\{x,x^R\},\{y,y^R\}\}) \setminus \{\{x,y\}\}$.

Case 2. $\mathcal{P} = \mathcal{P}^L \cup \mathcal{P}^R$.

(2.1) $|\mathcal{P}^L| < 2n-3$. Lemma 3.1 guarantees the existence of a free edge $\{x,y\} \in Q_n^L$. By the induction hypothesis there exist Hamiltonian cycles C in Q_n^L and C' in Q_n^R passing through $\mathcal{P}^L \cup \{x,y\}$ and $\mathcal{P}^R \cup \{x^R,y^R\}$, respectively. The desired Hamiltonian cycle of Q_{n+1} is then induced by edges of $(E(C) \cup E(C') \cup \{\{x,x^R\},\{y,y^R\}\}) \setminus \{\{x,y\},\{x^R,y^R\}\}$.

(2.2) $|\mathcal{P}^L| = 2n-3$ and $|\mathcal{P}^R| \leq 2$. First apply the induction to find a Hamiltonian cycle C in Q_n^L passing through $\mathcal{P}^L$. Since $|E(C)| - (|\mathcal{P}^L| + |\mathcal{P}^R|) \geq 2^n - (2n-1) \geq 3$ for $n \geq 3$, there has to be a free edge $\{x,y\} \in E(C) \setminus \mathcal{P}^L$. It remains to use the induction again to obtain a Hamiltonian cycle C' in Q_n^R passing through $\mathcal{P}^R \cup \{\{x^R,y^R\}\}$. The desired Hamiltonian cycle of Q_{n+1} is induced by edges of $(E(C) \cup E(C') \cup \{\{x,x^R\},\{y,y^R\}\}) \setminus \{\{x,y\},\{x^R,y^R\}\}$.

(2.3) $|\mathcal{P}^L| = 2n-2$ and $|\mathcal{P}^R| \leq 1$. First choose an edge $\{u,v\} \in \mathcal{P}^L$ such that v is an endvertex of a path of $\langle \mathcal{P}^L \rangle$. Next apply the induction to find in Q_n^L a Hamiltonian cycle C spanning $\mathcal{P}^L \setminus \{\{u,v\}\}$. If $\{u,v\} \in E(C)$, use the same construction as in case (2.2) to obtain the desired Hamiltonian cycle. If this is not the case, choose on C neighbors r and s of u and v, respectively, such that

(i) $\{u,r\} \notin \mathcal{P}^L$;

(ii) each of the two paths between r and s on C contains either u or v.

The way v has been chosen implies that $\{v,s\} \notin \mathcal{P}^L$. The existence of a path of length three between r and s means that $d(r,s)$ is odd and the same has to be true about $d(r^R,s^R)$. If $\{r^R,s^R\} \notin \mathcal{P}^R$, find a Hamiltonian path P in Q_n^R between r^R and s^R passing through $\mathcal{P}^R$, using Corollary 3.4 in case $|\mathcal{P}^R| = 1$ or Lemma 3.2 in case $\mathcal{P}^R = \emptyset$. The desired Hamiltonian cycle of Q_{n+1} is then induced by edges of

$(E(C) \cup E(P) \cup \{\{u,v\},\{r,r^R\},\{s,s^R\}\}) \setminus \{\{u,r\},\{v,s\}\}$.

If, however, $\mathcal{P}^R = \{\{r^R, s^R\}\}$, which means that $d(r,s) = 1$, choose an edge $\{p,q\} \in E(C) \setminus (\mathcal{P}^L \cup \{\{u,r\},\{v,s\}\})$. Note that as $|E(C)| - (|\mathcal{P}^L \setminus \{\{u,v\}\}| + 2) = 2^n - (2n-1) \geq 3$ for $n \geq 3$, this is always possible. Then use Corollary 3.4 to obtain a Hamiltonian path P in Q_n^R between p^R and q^R, passing through edge $\{r^R, s^R\}$. The desired Hamiltonian cycle of Q_{n+1} is then induced by edges of $(E(C) \cup E(P) \cup \{\{u,v\},\{r,s\},\{p,p^R\},\{q,q^R\}\}) \setminus \{\{u,r\},\{v,s\},\{p,q\}\}$.

(2.4) $|\mathcal{P}^L| = 2n-1$ and $|\mathcal{P}^R| = 0$. First choose two distinct edges $\{x,y\},\{u,v\} \in \mathcal{P}$ in the following way: If $\mathcal{P}$ forms a matching, the choice is arbitrary; otherwise x and v should be distinct endvertices of the same path, forming a connected component of $\langle \mathcal{P} \rangle$.

Next apply the induction to find in Q_n^L a Hamiltonian cycle C passing through $\mathcal{P} \setminus \{\{x,y\},\{u,v\}\}$. If at least one of the two chosen edges is included in C, use a construction from the preceding cases. If this is not the case, observe that the way $\{x,y\}$ and $\{u,v\}$ were chosen guarantees that we can assume without a loss of generality that one of the paths on C between x and v contains both y and u. Now consider the following subcases:

(2.4.1) $d_C(x,v) = 1$. Let y' and u' be the respective neighbors of y and u such that one of the paths between them on C contains x and v and the other contains y and u. Note that then $\{\{x,v\},\{y,y'\},\{u,u'\}\} \cap \mathcal{P} = \emptyset$. It remains to observe that $\langle (E(C) \cup \{\{x,y\},\{u,v\}\}) \setminus \{\{x,v\},\{y,y'\},\{u,u'\}\} \rangle$ forms a Hamiltonian path of Q_n^L passing through $\mathcal{P}$ and to apply Lemma 3.5.

(2.4.2) $d_C(x,v) \geq 2$. Choose neighbors x', y', u', v' of x, y, u, v on C, respectively, in the way described below. Note that in each case the four chosen vertices are pairwise distinct and, moreover, $\{\{x,x'\},\{y,y'\},\{u,u'\},\{v,v'\}\} \cap \mathcal{P} = \emptyset$.

(2.4.2.1) If one of the paths on C between x and y contains both u and v, choose the neighbors in such a way that

(i) one of the paths on C between x and y contains x' and y' and and the other contains u and v;

(ii) one of the paths on C between u and v contains x, y, v' and the other contains only u'.

(2.4.2.2) If one of the paths on C between x and y contains u and the other contains v, we claim that

$$\max(d_C(u,x), d_C(x,v), d_C(v,y)) \geq 3.$$

Indeed, if not, then $d_C(x,v) = 2$, and hence there is a common neighbor z of x and v on C. Now u, v, z, x forms a path of length three between u and x. It follows that $d(x,u)$ must be odd and our assumption $d_C(u,x) < 3$ implies that actually x and u are adjacent. Then apply the same argument to y and v to see that $\{y,v\} \in E(Q_n)$, too. Finally, observe that now x and v have three distinct neighbors in common, namely u, y, and z, which contradicts the $(0,2)$-property of the hypercube.

(2.4.2.2.1) If $d_C(u,x) \geq 3$, choose the neighbors in such a way that

(i) one of the paths on C between x and u contains x' and u' and the other y and v;

(ii) one of the paths on C between y and v contains all of y', v', x, u.

(2.4.2.2.2) If $d_C(x,v) \geq 3$, choose the neighbors in such a way that

(i) one of the paths on C between x and v contains x' and v' and the other u and y;

(ii) one of the paths on C between y and u contains all of y', u', x, v.

(2.4.2.2.3) $d_C(v, y) \geq 3$. This case is isomorphic to (2.4.2.2.1).
Now observe that in all cases described above,

$$\langle (E(C) \cup \{\{x, y\}, \{u, v\}\}) \setminus \{\{x, x'\}, \{y, y'\}, \{u, u'\}, \{v, v'\}\} \rangle$$

consists of two paths P_1 and P_2 such that

(i) $V(Q_n^L) = V(P_1) \dot{\cup} V(P_2)$;
(ii) $\mathcal{P} \subseteq E(P_1) \cup E(P_2)$;
(iii) endvertices of P_1 and P_2 are in $\{x', y', u', v'\}$.

It remains to note that the existence of a path x', x, y, y' means that $d(x', y')$ is odd and to apply Lemma 3.5. □

5. Concluding remarks. The existence of Hamiltonian cycles passing through prescribed edges is apparently related to the problem of Hamiltonicity of graphs with faulty edges mentioned in the introduction. Indeed, if two prescribed edges are incident with the same vertex v, any cycle passing through them must avoid all the remaining edges, incident with v. In particular, this argument can be used to show how Theorem 4.1 implies a classical result of [3] on the existence of Hamiltonian cycles of Q_n avoiding $n - 2$ forbidden edges (see [4] for details).

In may be of interest to explore further this connection comparing the complexity of both problems. The problem of Hamiltonicity of hypercubes with faulty edges is known to be NP-complete [2]. Does a similar result hold for the variant with prescribed edges?

Another generalization would lead to considering more than $2n - 3$ prescribed edges in Q_n and looking for conditions sufficient for the existence of a Hamiltonian cycle. A natural way to strengthen the condition of Theorem 4.1 is to consider only prescribed edges forming a matching. There is a related conjecture of Kreweras saying that *any perfect matching of Q_n ($n \geq 2$) is contained in a Hamiltonian cycle*. It can be easily seen to be true for $n = 2, 3$, while the case $n = 4$ was verified in [9]. To the best of our knowledge, no other results have been published.

Acknowledgments. I am very grateful to the anonymous referees for useful comments and critical remarks that helped to improve the exposition. I am also obliged to Rosťa Caha, Petr Gregor, and Vašek Koubek for fruitful discussions on this topic.

REFERENCES

[1] R. Caha and V. Koubek, *Hamiltonian cycles and paths with a prescribed set of edges in hypercubes and dense sets*, J. Graph Theory, submitted.
[2] M. Y. Chan and S.-J. Lee, *On the existence of Hamiltonian circuits in faulty hypercubes*, SIAM J. Discrete Math., 4 (1991), pp. 511–527.
[3] M.-S. Chen and K. G. Shin, *Processor allocation in an N-cube multiprocessor using Gray codes*, IEEE Trans. Comput., C-36 (1987), pp. 1396–1407.
[4] T. Dvořák and P. Gregor, *Hamiltonian paths with prescribed edges in hypercubes*, in Proceedings of Eurocomb '03 (European Conference on Combinatorics, Graph Theory and Applications, Prague, September 8–12, 2003), submitted.
[5] L. Gros, *Théorie du Baguenodier*, Aimé Vingtrinier, Lyon, 1872.
[6] F. Harary, *Graph Theory*, Addison-Wesley, New York, 1969.
[7] F. Harary, J. P. Hayes, and H.-J. Wu, *A survey of the theory of hypercube graphs*, Comput. Math. Appl., 15 (1988), pp. 277–289.
[8] I. Havel, *On Hamiltonian circuits and spanning trees of hypercubes*, Čas. Pěst. Mat., 109 (1984), pp. 135–152.

[9] G. Kreweras, *Matchings and Hamiltonian cycles on hypercubes*, Bull. Inst. Combin. Appl., 16 (1996), pp. 87–91.

[10] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann, San Mateo, CA, 1992.

[11] C. Savage, *A survey of combinatorial Gray codes*, SIAM Rev., 39 (1997), pp. 605–629.

[12] A. Sengupta, *On ring embedding in hypercubes with faulty nodes and links*, Inform. Process. Lett., 68 (1998), pp. 207–214.

SIAM J. Discrete Math.
Vol. 19, No. 1, pp. 145–148

ON PROJECTIVE CODES SATISFYING THE CHAIN CONDITION[*]

SYLVIA ENCHEVA[†]

Abstract. Projective codes with length above the Griesmer bound that satisfy the chain condition are discussed. Necessary and sufficient conditions for binary linear codes for which the chain condition holds are derived.

Key words. binary linear projective codes, chain condition

AMS subject classification. 94B05

DOI. 10.1137/S089548019631431X

1. Introduction.

In their paper on wire-tap channel II [4], Ozarow and Wyner discussed the use of linear codes for protecting information from intruders. It has been shown in [5] that the weight hierarchy of a linear code completely characterizes the performance of the code on the wire-tap channel. In addition, it has been pointed out that, for a related cryptographical application, the weight hierarchy also completely characterizes the performance of a linear code as a t-resilient function [1].

One motivation for studying the chain condition (CC) is that when two codes A and B satisfy the CC, the weight hierarchy of their product $A \otimes B$ can often be expressed in terms of those of A and B. Another motivation is related to the minimum trellis structure. A sufficient condition for optimal bit ordering in terms of a related chain formulation is stated in [3].

An $[n,k]$ code C is called *projective* if any two of its coordinates are linearly independent, i.e., if the dual code $C^\perp$ has minimum distance $d^\perp \geq 3$. The support of a binary vector is the set of its nonzero coordinates. The minimum support weight, d_r, of a code C is the size of the smallest support of any r-dimensional subcode of C. In particular $d_1 = d$. The *weight hierarchy* of C is $\{d_1, d_2, \ldots, d_k\}$.

The concept of the CC was introduced by Wei and Yang [6].

Definition 1.1. *An $[n,k]$ code C satisfies the CC if it is equivalent to a code $\tilde{C}$ such that there exists a chain of subcodes of $\tilde{C}$, $D_1 \subset D_2 \subset \cdots \subset D_k = \tilde{C}$, where, for $1 \leq r \leq k$, we have* $\dim(D_r) = r$ *and* $\chi(D_r) = \{1, 2, \ldots, d_r\}$.

Theorem 1.2 (see [2]). *A code C of length $n = g(k,d) + 1$ satisfies the CC.*

Theorem 1.3 (duality; see [5]). *Let C be an $[n,k]$ code and $C^\perp$ be its dual code. Then* $\{d_r : 1 \leq r \leq k\} = \{1, 2, \ldots, n\} \setminus \{n + 1 - d_r^\perp : 1 \leq r \leq n - k\}$.

From now on C is a binary linear projective code. In the rest of this paper we study conditions under which codes of length above the Griesmer bound satisfy the CC.

2. Main results.

For the description of generator matrices of projective codes we need some further notation. A column in this description represents a sequence of columns in the generator matrix, where a $(\star)$ can be either a 0 or a 1. The number of columns in the sequence is written above the column and $a + b + \cdots + z = l$. From now on G_n will be a generator matrix of a code C, written in the form

[*] Received by the editors December 27, 1996; accepted for publication (in revised form) October 29, 2004; published electronically June 16, 2005.
http://www.siam.org/journals/sidma/19-1/31431.html

[†] Stord/Haugesund University College, Bjørnsonsg. 45, 5528 Haugesund, Norway (sbe@hsh.no).

$$G_n = \left(\begin{array}{cccccc} & \overbrace{0\dots0}^{a} & \overbrace{0\dots0}^{b} & \dots & \overbrace{0\dots0}^{v} & \overbrace{0\dots0}^{z} \\ G_{n-l} & \dots & \dots & \dots & 0\dots0 & 0\dots0 \\ & 0\dots0 & 0\dots0 & \dots & 0\dots0 & 0\dots0 \\ \star\dots\star & 1\dots1 & 0\dots0 & \dots & 0\dots0 & 0\dots0 \\ \star\dots\star & \star\dots\star & 1\dots1 & \dots & 0\dots0 & 0\dots0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \star\dots\star & \star\dots\star & \star\dots\star & \dots & 0\dots0 & 0\dots0 \\ \star\dots\star & \star\dots\star & \star\dots\star & \dots & 1\dots1 & 0\dots0 \\ \star\dots\star & \star\dots\star & \star\dots\star & \dots & \star\dots\star & 1\dots1 \end{array} \right), \tag{2.1}$$

where G_{n-l} generates an $[n-l, k-l+1]$ code.

For given k and d, there exists an $[n,k,d]$ code only if

$$n(k,d) \geq g(k,d) := \sum_{i=0}^{k-1} \left\lceil \frac{d}{2^i} \right\rceil,$$

where $\lceil x \rceil$ is the least integer not smaller than x; this is known as the Griesmer bound.

THEOREM 2.1. *If C is an $[n,k]$ code with dual distance at least 3 and generator matrix G_n, where G_{n-l} generates an $[n-l, k-l+1]$ code that satisfies the CC, then C satisfies the CC.*

Proof. Since C is projective, its generator matrix can be written in the form (2.1), where G_{n-l} spans an $[n-l, k-l+1]$ code C_{k-l+1}.

Since C_{k-l+1} satisfies the CC we conclude that C does also. □

Example 1. Let $C_{11}[33,11,11]$ be the dual code of a cyclic $[33,22,6]$ code with generator polynomial 101001100101. C_{11} is a projective code since $d_1^{\perp} = 6$. A generator matrix $G_{11\times33}$ of C_{11} may be written in the form

$$G_{11\times33} = (I_{11\times11} | A_{11\times22}),$$

where $I_{11\times11}$ is the identity matrix. $A_{11\times22}$ contains columns of weight 5. By deleting all rows in $G_{11\times33}$ that have support size 1 in a column of weight 5 in $A_{11\times22}$, we obtain a $[27,6,11]$ code with $d_1^{\perp} = 3$. We consider its generator matrix, written in the form

$$G_{6\times27} = (I_{6\times6} | A_{6\times21}),$$

where $I_{6\times6}$ is the identity matrix. $A_{6\times21}$ contains columns of weight 2. By deleting the two rows in $G_{6\times27}$ that have support size 1 in a column of weight 2 in $A_{6\times21}$, we obtain a $[24,4,12]$ code. The code $[24,4,12]$ satisfies the CC since $24 = g(4,12)+1$. Then by Theorem 2.1 the $[27,6,11]$ code and the $[33,11,11]$ code will satisfy the CC.

PROPOSITION 2.2. *Let C_1 be an $[n_1 = g(k,d)+3, k_1, d_1]$ projective code with generator matrix G_{n_1}. If there exists a code $C[n_1-1, k, d \leq 2^{k-2}]$ with generator matrix G_{n_1-1} such that $d_1^{\perp} = 3$, $d_2^{\perp} = 5$, then C_1 satisfies the CC.*

Proof. According to Theorem 2.1 all we need to prove is that C satisfies the CC.

From Theorem 1.3 and conditions $d_1^{\perp} = 3$, $d_2^{\perp} = 5$, it follows that the weight hierarchy of C is $d_i = g(i,d)$ for $1 \leq i \leq k-3$, $d_{k-2} = g(k-2,d)+1$, and $d_i = g(i,d)+2$ for $i = k-1, k$.

Since $d \leq 2^{k-2}$, $d_{k-2} = g(k-2,d)+1$, and $d_2^{\perp} = 5$, we have $d_{k-3} = g(k-3,d)$. By Theorem 1.3, the highest d_i's are $d_k = n$, $d_{k-1} = n-1$, $d_{k-2} = n-3$, $d_{k-3} = n-5$. Since $d_1^{\perp} = 3$, a generator matrix of C can be written in the form (2.1) with

$n-l=k-2$, $a=2$, where G_{k-2} generates an $[n-3=g(k-2,d)+1,k-2,d]$ code C_{k-2}. The code C_{k-2} generated by the first $k-2$ rows has length $n-3=g(k-2,d)+1$ and, according to Theorem 1.2, satisfies the CC. From $d_{k-3}=g(k-3,d)$ it follows that the weight hierarchy of C_{k-2} coincides with the first $k-2$ elements of the weight hierarchy of C, completing the proof. □

DEFINITION 2.3. *Let c be a nonzero codeword of C and G a generator matrix for C, with c as the first row. The code generated by the restriction of G to the columns in which c has zero coordinates is called the residual code of C with respect to c and is denoted by $Res(C,c)$.*

THEOREM 2.4. *A necessary and sufficient condition for an $[n=g(k,d)+\alpha,k,d]$ code C with weight hierarchy $\{d_1=d,d_2,\ldots,d_k\}$ to satisfy the CC is that there exists an $[n-d,k-1,d'_1=d_2-d]$ code $C'=Res(C,c)$ with weight hierarchy $\{d'_1,d'_2=d_3-d,\ldots,d'_{k-1}=d_k-d\}$, which satisfies the CC.*

Proof. Let C be a code as in the theorem and which satisfies the CC. Without loss of generality its generator matrix G can be written in the form (2.1) in such a way that its rows display the chain of subcodes of Definition 1.1. Let c be the first row in G. We will prove that $C'=Res(C,c)$ satisfies the CC for $\alpha=2$. The proof for $\alpha>2$ is similar.

Since C is a code with length 2 above the Griesmer bound, there are integers $0<m\leq p<k$ such that $d_1=d,d_2=g(2,d),\ldots,d_m=g(m,d),d_{m+1}=g(m+1,d)+1$, $\ldots,d_p=g(p,d)+1,d_{p+1}=g(p+1,d)+2,\ldots,d_k$. Now, we prove that $m\neq p$. Suppose that there exists a code C with parameters as described in the theorem and an integer s such that $0<s<k$, $B_j=0$, $j=1,2,\ldots,k-s-2$, and $B_{k-s-1}>0$ with, furthermore, $d_i=g(i,d)$ for $1\leq i\leq s$, $d_i=g(i,d)+2$ for $s+1\leq i\leq k$. In other words, suppose that $l=p=s$. Assume w.l.o.g. that a codeword of $C^{\perp}$ has for support the last $k-s$ positions. Then a generator matrix of C can be written in the form (2.1), where G_s generates a subcode of length $n-(k-s)$ and dimension s and where $a=3$.

The last $k-s-2$ columns in (2.1) are linear independent. The second column after G_s, (h_2), is a linear combination of all $k-s-2$ columns on its right since $B_{k-s-1}>0$. The first column after G_s differs from h_2 by the projectivity assumption. Therefore, it can be a linear combination of at most $k-s-3$ columns chosen among the last $k-s-2$. This contradicts the assumption that $B_j=0$, $j=1,2,\ldots,k-s-2$. Hence $m\neq p$.

Let us calculate d'_{l-1} and d'_l: $d'_{l-1}=d_l-d=g(l-1,d'_1)$, $d'_l=d_{l+1}-d=g(l,d'_1)+1$. The same relations are valid between d_p, d'_{p-1} and d_{p+1}, d'_p. Therefore, C' has weight hierarchy $d'_1=d_2-d,d'_2=d_3-d,\ldots,d'_{k-1}=d_k-d$ and satisfies the CC.

Conversely, let C be an $[n=g(k,d)+2,k,d]$ code with weight hierarchy $\{d_1=d$, $d_2,\ldots,d_k\}$ and such that there exists an $[n-d,k-1,d'_1=d_2-d]$ code $C'=Res(C,c)$ with weight hierarchy $\{d'_1,d'_2=d_3-d,\ldots,d'_{k-1}=d_k-d\}$ satisfying the CC. From Definition 2.3 it follows that C satisfies the CC. □

REFERENCES

[1] B. CHOR, O. GOLDREICH, J. FRIEDMAN, S. RUDICH, AND R. SMOLESKY, *The bit extraction problem of t-resilient functions*, in Proceedings of the 26th Annual IEEE Symposium on Foundations of Computer Science, IEEE Press, Piscataway, NJ, 1985, pp. 396–407.

[2] T. HELLESETH, T. KLØVE, AND Ø. YTREHUS, *Generalized Hamming weights of linear codes*, IEEE Trans. Inform. Theory, 38 (1992), pp. 1133–1140.

[3] T. KASAMI, T. TAKATA, T. FUJIWARA, AND S. LIN, *On the optimum bit orders with respect to the state complexity of trellis diagrams for binary linear codes*, IEEE Trans. Inform. Theory, 39 (1993), pp. 242–243.

[4] L. H. Ozarow and A. D. Wyner, *Wire-tap channel* II, AT&T Bell Labs. Tech. J., 63 (1984), pp. 2135–2157.

[5] V. K. Wei, *Generalized Hamming weights for linear codes*, IEEE Trans. Inform. Theory, 37 (1991), pp. 1412–1418.

[6] V. K. Wei and K. Yang, *On the generalized Hamming weights of product codes*, IEEE Trans. Inform. Theory, 39 (1993), pp. 1709–1713.

SIAM J. DISCRETE MATH.
Vol. 19, No. 1, pp. 149–164

A RANDOM VERSION OF SHEPP'S URN SCHEME*

ROBERT W. CHEN[†], ALAN ZAME[†], CHIEN-TAI LIN[‡], AND HSIU-FEN WU[‡]

Abstract. In this paper, we consider the following random version of Shepp's urn scheme: A player is given an urn with n balls. p of these balls have value $+1$ and $n-p$ have value -1. The player is allowed to draw balls randomly, without replacement, until he or she wants to stop. The player knows n, the total number of balls, but knows only that p, the number of balls of value $+1$, is a number selected randomly from the set $\{0, 1, 2, \ldots, n\}$. The player wishes to maximize the expected value of the sum of the balls drawn. We first derive the player's optimal drawing policy and an algorithm to compute the player's expected value at the stopping time when he or she uses the optimal drawing policy. Since the optimal drawing policy is rather intricate and the computation of the player's optimal expected value is quite cumbersome, we present a very simple drawing policy, which is asymptotically optimal. We also show that this random urn scheme is equivalent to a random coin tossing problem.

Key words. urn scheme, optimal drawing policy, random coin tossing process, stopping time, the "k" in the hole policy

AMS subject classifications. Primary, 60G40; Secondary, 60K99

DOI. 10.1137/S0895480102418099

1. Introduction.

In [8], Shepp considered the following optimal stopping problem: A player is given an urn with n balls. p of these balls have value $+1$ and $n-p$ have value -1. The player knows n and p. The player's goal is to maximize the expected value of the sum of the balls drawn. The player may draw as long as he or she wishes, without replacement. Shepp was interested in knowing for what n and p there is a drawing policy for which $V(n,p)$, the expected value of the game if there are n balls, p of which are $+1$, is positive. He showed that for a given n there is an integer $\gamma(n)$ such that $V(n,p) > 0$ if and only if $p \geq \gamma(n)$. More precisely, he showed that there exists a $\beta(p)$ for which $V(n,p) > 0$ if and only if $0 \leq n-p \leq \beta(p)$.

In [1], Boyce was interested in the following bond-selling problem: A corporation must repay 10 million dollars in bank loans in three months, and it wishes to sell bonds to repay the loan. However, the company's economists predict that in three months bond prices will be lower (interest rates higher). Should the corporation issue the bonds now, wait a month or two, or wait the full three months? For this bond-selling problem, Boyce introduced a random version of Shepp's urn scheme, which can be stated as follows: A player is given an urn with n balls. p of these balls have value $+1$ and $n-p$ have value -1. The player is allowed to draw balls randomly, without replacement, until he or she wants to stop. The player knows n, the total number of balls, but only knows the distribution of p, the number of balls of value $+1$. The player wishes to maximize the expected value of the sum of his draws. Boyce briefly studied this problem and proposed a procedure to compute the player's expected value at the stopping time when he or she uses an optimal drawing policy.

*Received by the editors November 18, 2002; accepted for publication (in revised form) June 26, 2004; published electronically June 16, 2005.

http://www.siam.org/journals/sidma/19-1/41809.html

[†]Department of Mathematics, University of Miami, Coral Gables, FL 33124 (chen@math.miami.edu, zame@math.miami.edu). The work of the first author was partially supported by the Department of Mathematics, Tamkang University of Taiwan, and was partially done during his visit.

[‡]Department of Mathematics, Tamkang University, Taipei, Taiwan (chien@math.tku.edu.tw, hfwu@math.tku.edu.tw).

In this paper, we study this random version of Shepp's urn scheme for the case when the distribution of p is uniform over the set $\{0, 1, 2, \dots, n\}$. In section 2, we derive the player's optimal drawing policy and an algorithm to compute the player's expected value at the stopping time when he or she uses the optimal drawing policy. It will be seen that the optimal drawing policy is very intricate. Also the computation of the player's optimal expected value at the stopping time is quite cumbersome, especially as n gets large. In section 3, we present a very simple drawing policy and show that this simple drawing policy is not only asymptotically optimal, but also performs very well even when n is small. Our data reveal that for $n = 10, 20, \dots, 1{,}000$, the difference between the expected value at the stopping time under an optimal drawing policy and the expected value at the stopping time under this simple drawing policy is less than 1. In section 4, we will show that this random urn problem can be stated as a random coin tossing problem.

2. An optimal drawing policy. In order to compute the expected value of the game, we consider what the remaining value of the game would be, conditioned on the outcome of the first k draws. Suppose there are n balls and k balls have been drawn, j of which have value +1. Let $E(n, k, j)$ denote the remaining expected value of the game from this point on. The following lemma gives the critical recursion for E. Its proof is a straightforward application of Bayes' law, but we include a proof for the sake of completeness.

LEMMA 1. *If* $0 \le j \le k \le n-1$, *then*

$$E(n,k,j) = \max\left\{0, \frac{2j-k}{k+2} + \frac{j+1}{k+2}E(n,k+1,j+1) + \frac{k-j+1}{k+2}E(n,k+1,j)\right\}.$$

Proof. Having drawn k balls, with j "+1 balls," the player has to decide whether to play any further. The player may draw another ball; suppose that the conditional probability that it is +1 is $\alpha(n, k, j) = \alpha$ and that it is -1 is $\beta = 1 - \alpha$. The expected value of the remainder of the game, if another ball is drawn, is then

$$\alpha - \beta + \alpha E(n, k+1, j+1) + \beta E(n, k+1, j).$$

Thus, the player should draw another ball if this is positive and stop otherwise. The recursion relation will then follow if we can show that $\alpha = \frac{j+1}{k+2}$. To do this, let $X_i = 1$ if the ith draw is a +1, or let $X_i = 0$ otherwise for all $i = 1, 2, \dots, k+1$. Let $S_k = \sum_{i=1}^{k} X_i$. Then it is easy to see that

$$\alpha = P(X_{k+1} = 1 | S_k = j) = P([S_k = j] \cap [X_{k+1} = 1])/P(S_k = j).$$

For each $i = 0, 1, 2, \dots, n$, let A_i denote the event $[p = i]$. Since the distribution of p is uniform over the set $\{0, 1, 2, \dots, n\}$, $P(A_i) = \frac{1}{n+1}$ for all $i = 0, 1, 2, \dots, n$.

$$\begin{aligned} P(S_k = j) &= \sum_{i=1}^{n} P([S_k = j] \cap A_i) = \sum_{i=1}^{n} P([S_k = j] | A_i) P(A_i) \\ &= \frac{1}{n+1} \sum_{i=j}^{n-k+j} \frac{\binom{i}{j}\binom{n-i}{k-j}}{\binom{n}{k}} = \frac{\binom{n+1}{k+1}}{(n+1)\binom{n}{k}} = \frac{1}{k+1} \end{aligned}$$

since $P([S_k = j] | A_i) = 0$ if $i < j$ or $i > n - k + j$.

Similarly,

$$P([S_k = j] \cap [X_{k+1} = 1]) = \sum_{i=1}^{n} P([S_k = j] \cap [X_{k+1} = 1] \cap A_i)$$

$$= \sum_{i=1}^{n} P([S_k = j] \cap [X_{k+1} = 1] | A_i) P(A_i) = \frac{1}{n+1} \sum_{i=j+1}^{n-k+j} \frac{(i-j)\binom{i}{j}\binom{n-i}{k-j}}{(n-k)\binom{n}{k}}$$

$$= \frac{(j+1)}{(n+1)(n-k)\binom{n}{k}} \sum_{i=j+1}^{n-k+j} \binom{i}{j+1}\binom{n-i}{k-j} = \frac{(j+1)}{(k+1)(k+2)}$$

since $P([S_k = j] \cap [X_{k+1} = 1] | A_i) = 0$ if $i < j+1$ or $i > n-k+j$.

Therefore,

$$\alpha = \left\{ \frac{(j+1)}{(k+1)(k+2)} \right\} \Big/ \left\{ \frac{1}{(k+1)} \right\} = \frac{(j+1)}{(k+2)}.$$

This completes the proof of Lemma 1. □

It is clear that $E(n,n,j) = 0$ for all $j = 0, 1, 2, \ldots, n$ since there are no balls left. It is also clear that if the player draws k balls, j of which have value $+1$, the player should stop drawing unless $E(n,k,j) > 0$. Therefore, the optimal drawing policy can be stated as follows: At the beginning, the player will draw a ball if and only if $E(n,0,0) > 0$. Suppose that the player has drawn k balls, j of which have value $+1$; the player will continue to draw if and only if $E(n,k,j) > 0$.

Boyce briefly studied this problem in [1]. He produces a procedure for computing $E(n,0,0)$. The procedure requires computing all of $E(n,k,j)$ for $0 \le j \le k < n$ in order to get $E(n,0,0)$. It is clear that the computation is very cumbersome, and for each new n, we have to compute all new $E(n,k,j)$ for all $0 \le j \le k < n$ to determine the new optimal drawing policy.

Table 1 gives partial values of $E(120,k,j)$ for $0 \le j \le k \le 25$.

The optimal drawing policy can also be stated as follows: For each given n, we create a table such as Table 1. We start from the position in which $k = 0$ and $j = 0$ and move one step down or one step to the southeast according to when a "-1" ball is drawn or a "$+1$" ball is drawn. We will stop drawing if and only if we reach a zero. However, even when n is moderate, it takes too much time to construct such a table.

THEOREM 1. *$E(n,0,0)$ is a strictly increasing function of n.*

Proof. It is sufficient to show that $E(n+1,k,j) \ge E(n,k,j)$ and $E(n+1,k,k) > E(n,k,k)$ for all $0 \le j \le k \le n$. Since $E(n+1,n,j) = \max\{0, \frac{2j-n}{n+2}\}$ and $E(n,n,j) = 0$ for all $0 \le j \le n$, and since $E(n+1,k,j) = \max\{0, \frac{2j-k}{k+2} + \frac{j+1}{k+2}E(n,k+1,j+1) + \frac{k-j+1}{k+2}E(n,k+1,j)\}$ and $E(n,k,j) = \max\{0, \frac{2j-k}{k+2} + \frac{j+1}{k+2}E(n,k+1,j+1) + \frac{k-j+1}{k+2}E(n,k+1,j)\}$ for all $0 \le j \le k \le n-1$ and $n \ge 1$, by mathematical induction we can conclude that $E(n+1,k,k) > E(n,k,k)$ for all $0 \le k \le n$ and $n \ge 1$. Therefore, $E(n,0,0)$ is a strictly increasing function of n. □

THEOREM 2. *$E(n,0,0) \le \frac{n}{4} + o(n)$.*

Table 1
$E(120, k, j)$.

k	$j = 0$	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$	$j = 7$	$j = 8$
0	27.49	-	-	-	-	-	-	-	-
1	6.98	48.00	-	-	-	-	-	-	-
2	1.28	19.38	61.80	-	-	-	-	-	-
3	0	7.13	31.63	71.20	-	-	-	-	-
4	0	2.02	15.30	42.19	77.70	-	-	-	-
5	0	0.23	6.62	23.98	50.79	82.28	-	-	-
6	0	0	2.29	12.72	32.17	57.64	85.55	-	-
7	0	0	0.44	6.03	19.41	39.42	63.05	87.91	-
8	0	0	0	2.33	10.91	26.01	45.63	67.32	89.81
9	0	0	0	0.56	5.49	16.32	32.13	50.84	70.68
10	0	0	0	0	2.28	9.55	21.80	37.61	55.18
11	0	0	0	0	0.61	5.02	14.07	27.04	42.39
12	0	0	0	0	0	2.20	8.47	18.73	31.86
13	0	0	0	0	0	0.64	4.60	12.35	23.27
14	0	0	0	0	0	0.01	2.11	7.60	16.37
15	0	0	0	0	0	0	0.67	4.24	10.97
16	0	0	0	0	0	0	0.04	2.00	6.87
17	0	0	0	0	0	0	0	0.68	3.91
18	0	0	0	0	0	0	0	0.07	1.89
19	0	0	0	0	0	0	0	0	0.67
20	0	0	0	0	0	0	0	0	0.09
21	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0

Table 1
Continued.

k	$p = j$	$j = 10$	$j = 11$	$j = 12$	$j = 13$	$j = 14$	$j = 15$	$j = 16$	$j = 17$
0	-	-	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-
9	90.82	-	-	-	-	-	-	-	-
10	73.35	91.67	-	-	-	-	-	-	-
11	58.77	75.47	92.23	-	-	-	-	-	-
12	46.51	61.75	77.15	95.57	-	-	-	-	-
13	36.19	50.04	64.22	78.47	92.73	-	-	-	-
14	27.53	40.02	53.05	66.26	79.50	92.75	-	-	-
15	20.35	31.44	43.37	55.61	67.94	80.29	92.65	-	-
16	14.50	24.15	34.96	46.29	57.79	69.33	80.89	92.44	-
17	9.84	18.03	27.68	38.10	48.82	59.64	70.47	81.32	92.16
18	6.25	12.97	21.43	30.92	40.88	51.01	61.20	71.40	81.60
19	3.61	8.90	16.12	24.64	33.83	43.32	52.91	62.52	72.14
20	1.78	5.72	11.70	19.19	27.60	36.45	45.47	54.55	63.64
21	0.65	3.44	8.09	14.53	22.11	30.31	38.78	47.36	55.96
22	0.10	1.68	5.25	10.61	17.32	24.84	32.76	40.86	49.00
23	0	0.63	3.10	7.39	13.18	19.98	27.35	34.97	42.69
24	0	0.10	1.57	4.83	9.68	15.72	22.49	29.64	36.95
25	0	0	0.60	2.88	6.79	12.02	18.17	24.82	31.72

TABLE 1
Continued.

k	$j = 18$	$j = 19$	$j = 20$	$j = 21$	$j = 22$	$j = 23$	$j = 24$	$j = 25$
0	-	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-
18	91.80	-	-	-	-	-	-	-
19	81.76	91.38	-	-	-	-	-	-
20	72.73	81.82	90.91	-	-	-	-	-
21	64.57	73.17	81.78	90.39	-	-	-	-
22	57.17	65.33	73.50	81.67	89.83	-	-	-
23	50.44	58.20	65.96	73.72	81.48	89.24	-	-
24	44.31	51.69	59.08	66.46	73.85	81.23	88.62	-
25	38.72	45.74	52.77	59.81	66.85	73.89	80.93	87.96

Proof. Following Boyce [2], where the player starts with an urn with n balls and a known p of "+1" balls, let $V(n, p)$ be the player's expected score at the stopping time when he or she uses the optimal drawing policy. It is easy to see that $E(n, 0, 0) \leq \frac{1}{n+1} \sum_{j=0}^{n} V(n, j)$. If $n = 2m$, then by a theorem of Shepp, $V(2m, j) \leq V(2m, m)$ for all $j = 0, 1, 2, \ldots, m$ and $V(2m, j) \leq 2j - 2m + V(2m, m)$ for all $j = m + 1, m + 2, \ldots, 2m$. Since $V(2m, m) \approx \sqrt{m}$, it is easy to see that $E(n, 0, 0) \leq \frac{n}{4} + o(n)$. The proof for the case when n is odd is similar.

In theory, we can compute the expected score at the stopping time under the optimal drawing policy and describe the optimal drawing policy for each given positive integer n. However, even when n is just moderately large, the computation is very cumbersome and it is very difficult to describe the optimal drawing policy precisely. In section 3, we will present a simple drawing policy, which is not only asymptotically optimal, but also performs very well even when n is small. Our data reveal that for $n = 10$, $20, \ldots, 1{,}000$, $E(n, 0, 0) - W(n, k_n) < 1$, where $W(n, k_n)$ is the expected value at the stopping time when the player uses the simple drawing policy, which will be introduced in section 3. □

3. A simple drawing policy. One natural approach to determine when to stop is to play until we are a certain amount "in the hole." Here we continue drawing until the number of "-1" balls drawn is k more than the number of "+1" balls drawn. We will call this strategy "the k in the hole drawing policy." Let $W(n, k)$ be the expected value of the game following "the k in the hole drawing policy" when the urn originally contains n balls. One would expect the optimal choice for k to depend on n. We will show that it does. We will also show how to compute this optimal k very quickly. Most important, we will show that for any given n "the k in the hole drawing policy" is asymptotically optimal if we choose the best k.

THEOREM 3. *For each integer* $1 \le k \le n$, *if* $n-k$ *is even,*

$$W(n,k) = \frac{1}{n+1}\left\{\frac{(n-k+2)(n-k)}{4} - \sum_{j=(n-k)/2}^{n-k}(2j+k-n)\frac{\binom{n}{k+j}}{\binom{n}{j}}\right\},$$

and if $n-k$ *is odd,*

$$W(n,k) = \frac{1}{n+1}\left\{\frac{(n-k+1)^2}{4} - \sum_{j=(n-k+1)/2}^{n-k}(2j+k-n)\frac{\binom{n}{k+j}}{\binom{n}{j}}\right\}.$$

Proof. We give the proof when $n-k$ is even; when $n-k$ is odd the proof is similar. For each $j = 0, 1, 2, \ldots, n$, let $W(n,k,j)$ be the expected value at the stopping time when the player uses "the k in the hole drawing policy" and the urn originally contains $n-j$ balls of value -1 and j balls of value of $+1$. It is easy to see that

$$W(n,k) = \frac{1}{n+1}\sum_{j=0}^{n} W(n,k,j).$$

It is also clear that $W(n,k,j) = 2j-n$ if $j > n-k$ and $W(n,k,j) = -k$ if $j < (n-k)/2$. If $(n-k)/2 \le j \le n-k$, then by the reflection principle [5, p. 72], $W(n,k,j) = 2j-n$ with probability $1-\binom{n}{k+j}/\binom{n}{j}$ and $W(n,k,j) = -k$ with probability $\binom{n}{k+j}/\binom{n}{j}$. Therefore,

$$\begin{aligned}
W(n,k) &= \frac{1}{n+1}\sum_{j=0}^{n} W(n,k,j)\\
&= \frac{1}{n+1}\left\{\sum_{j=0}^{(n-k-2)/2}(-k) + \sum_{j=n-k+1}^{n}(2j-n)\right.\\
&\qquad\left. + \sum_{j=(n-k)/2}^{n-k}\left\{(2j-n)\left[1-\frac{\binom{n}{k+j}}{\binom{n}{j}}\right] + (-k)\frac{\binom{n}{k+j}}{\binom{n}{j}}\right\}\right\}\\
&= \frac{1}{n+1}\left\{-\frac{k(n-k)}{2} + \sum_{j=(n-k)/2}^{n}(2j-n) - \sum_{j=(n-k)/2}^{n-k}(2j+k-n)\frac{\binom{n}{k+j}}{\binom{n}{j}}\right\}\\
&= \frac{1}{n+1}\left\{\frac{(n-k+2)(n-k)}{4} - \sum_{j=(n-k)/2}^{n-k}(2j+k-n)\frac{\binom{n}{k+j}}{\binom{n}{j}}\right\}.
\end{aligned}$$

This completes the proof of Theorem 3. □

For each positive integer n, let

$$\begin{aligned}
k_n &= \min\{k \mid 1 \le k \le n,\ W(n,k) = \max\{W(n,j) \mid 1 \le j \le n\}\},\\
\underline{k}_n &= \max\{k \mid 1 \le k \le n,\ (n+1)(n-k)^2 \ge 8n(n-1)k^3\},\\
\overline{k}_n &= \min\{k \mid 1 \le k \le n,\ (n+1)(n+k)^2 \le 2(n-k)^2k^3\}.
\end{aligned}$$

TABLE 2
$W(n,k)$, $E(n,0,0)$, $\underline{k}_n$, k_n, and $\overline{k}_n$.

n	$W(n,1)$	$W(n,2)$	$W(n,3)$	$W(n,4)$	$W(n,5)$	$E(n,0,0)$	$\underline{k}_n$,	k_n	$\overline{k}_n$
10	**1.65**	1.59	1.33	1.03	0.77	**1.65**	1	1	3
20	3.57	**3.82**	3.61	3.29	2.95	**3.82**	2	2	3
30	5.50	**6.08**	5.96	5.66	5.31	**6.08**	2	2	3
40	7.43	**8.35**	8.33	8.06	7.71	**8.37**	2	2	4
50	9.36	10.62	**10.70**	10.47	10.14	**10.70**	2	3	4
60	11.29	12.89	**13.07**	12.89	12.57	**13.08**	2	3	4
70	13.22	15.16	**15.45**	15.31	15.01	**15.46**	3	3	4
80	15.15	17.43	**17.83**	17.73	17.45	**17.85**	3	3	4
90	17.08	19.71	**20.21**	20.15	19.90	**20.25**	3	3	4
100	19.01	21.98	**22.59**	22.58	22.34	**22.66**	3	3	4
200	38.32	44.71	46.41	**46.86**	46.85	**46.98**	3	4	5
300	57.64	67.45	70.24	71.15	**71.37**	**71.53**	4	5	6
400	76.95	90.19	94.07	95.44	95.90	**96.17**	4	6	6
500	96.27	112.94	117.91	119.74	120.43	**120.89**	4	6	7
600	115.58	135.68	141.74	144.04	144.96	**145.65**	5	7	7
700	134.90	158.42	165.57	168.33	169.50	**170.44**	5	7	8
800	154.21	181.16	189.40	192.63	194.03	**195.25**	5	7	8
900	173.53	203.90	213.23	216.93	218.56	**220.07**	5	8	8
1,000	192.84	226.64	237.07	241.23	243.10	**244.90**	5	8	9

TABLE 2
Continued.

n	$W(n,6)$	$W(n,7)$	$W(n,8)$	$W(n,9)$	$W(n,10)$	$E(n,0,0)$	$\underline{k}_n$,	k_n	$\overline{k}_n$
10	0.53	0.34	0.18	0.08	0	**1.65**	1	1	3
20	2.61	2.28	1.97	1.68	1.41	**3.82**	2	2	3
30	4.94	4.57	4.21	3.86	3.52	**6.08**	2	2	3
40	7.33	6.95	6.56	6.18	5.81	**8.37**	2	2	4
50	9.76	9.37	8.97	8.57	8.18	**10.70**	2	3	4
60	12.20	11.80	11.40	10.99	10.58	**13.08**	2	3	4
70	14.64	14.25	13.84	13.43	13.02	**15.46**	3	3	4
80	17.09	16.70	16.29	15.88	15.46	**17.85**	3	3	4
90	19.55	19.16	18.75	18.34	17.91	**20.25**	3	3	4
100	22.01	21.62	21.22	20.80	20.37	**22.66**	3	3	4
200	46.63	46.32	45.95	45.55	45.13	**46.98**	3	4	5
300	71.29	71.05	70.73	70.36	69.96	**71.53**	4	5	6
400	**95.95**	95.80	95.53	95.20	94.82	**96.17**	4	6	6
500	**120.62**	120.54	120.33	120.04	119.69	**120.89**	4	6	7
600	145.28	**145.30**	145.14	144.88	144.55	**145.65**	5	7	7
700	169.95	**170.05**	169.94	169.72	169.43	**170.44**	5	7	8
800	194.42	**194.80**	194.75	194.57	194.30	**195.25**	5	7	8
900	219.29	219.55	**219.56**	219.41	219.17	**220.07**	5	8	8
1,000	243.96	244.30	**244.37**	244.26	244.05	**244.90**	5	8	9

It is easy to see that $\underline{k}_n \leq \overline{k}_n$ for each positive integer n. In Theorem 8, we will prove that $\underline{k}_n \leq k_n \leq \overline{k}_n$ for each positive integer n.

Table 2 provides some numerical values of $W(n,k)$, $E(n,0,0)$, k_n, $\underline{k}_n$, and $\overline{k}_n$ for various n and k.

Table 2 provides numerical evidence of the following: (I) For each k, $W(n,k)$ is increasing in n. (II) For each n, $W(n,k)$ first increases and then decreases in k (for small n, $W(n,k)$ is decreasing in k). (III) $\underline{k}_n \leq k_n \leq \overline{k}_n$. (IV) $W(n,k_n) \approx \frac{n}{4}$.

Theorem 4 proves (I). Theorem 5 proves (IV). Theorem 7 gives a partial answer to (II). Theorem 8 proves (III).

THEOREM 4. *For each fixed k $(1 \leq k \leq n)$, $W(n,k)$ is increasing in n.*

Proof. We will prove the case when $n-k$ is even since the proof for the case when $n-k$ is odd is similar. By Theorem 3

$$W(n+1,k) = \frac{1}{n+2}\left\{\frac{(n-k+2)^2}{4} - \sum_{j=(n-k+2)/2}^{n-k+1}(2j+k-n-1)\frac{\binom{n+1}{k+j}}{\binom{n+1}{j}}\right\}$$
$$= \frac{1}{n+2}\left\{\frac{(n-k+2)^2}{4} - \sum_{j=(n-k)/2}^{n-k}(2j+k+1-n)\frac{\binom{n+1}{k+j+1}}{\binom{n+1}{j+1}}\right\}$$

and

$$W(n,k) = \frac{1}{n+1}\left\{\frac{(n-k+2)(n-k)}{4} - \sum_{j=(n-k)/2}^{n-k}(2j+k-n)\frac{\binom{n}{k+j}}{\binom{n}{j}}\right\}.$$

Therefore,

$$W(n+1,k) - W(n,k) = \frac{1}{(n+1)(n+2)}\left\{\frac{(n+2)^2-k^2}{4}\right.$$

$$\left.-\sum_{j=(n-k)/2}^{n-k}\left\{(n+1)(2j+k+1-n)\frac{\binom{n+1}{k+j+1}}{\binom{n+1}{j+1}} - (n+2)(2j+k-n)\frac{\binom{n}{k+j}}{\binom{n}{j}}\right\}\right\}.$$

To show that $W(n+1,k) - W(n,k) \geq 0$, it is sufficient to show that

$$\sum_{j=(n-k)/2}^{n-k}\left\{(n+1)(2j+k+1-n)\frac{\binom{n+1}{k+j+1}}{\binom{n+1}{j+1}} - (n+2)(2j+k-n)\frac{\binom{n}{k+j}}{\binom{n}{j}}\right\}$$
$$\leq \frac{(n+2)^2-k^2}{4}.$$

Notice that

$$\sum_{j=(n-k)/2}^{n-k}\left\{(n+1)(2j+k+1-n)\frac{\binom{n+1}{k+j+1}}{\binom{n+1}{j+1}} - (n+2)(2j+k-n)\frac{\binom{n}{k+j}}{\binom{n}{j}}\right\}$$

$$= \sum_{j=(n-k)/2}^{n-k}\left\{\frac{(n+1)(2j+k+1-n)(j+1)-(n+2)(2j+k-n)(k+j+1)}{k+j+1}\right\}\frac{\binom{n}{k+j}}{\binom{n}{j}}.$$

For fixed $1 \leq k \leq n$, let

$$g(j) = (n+1)(2j+k+1-n)(j+1) - (n+2)(2j+k-n)(k+j+1).$$

After simplification,

$$g(j) = -2j^2 - (2nk+5k+1-2n)j + (n^2k+2nk+2n+1-nk^2-2k^2-k).$$

Since $k \geq 1$, $g(j)$ is decreasing for $j \geq (n-k)/2$ and $g(j) \leq 0$ if

$$j \geq \frac{1}{4}\{\ 2n - 2nk - 5k - 1 + \sqrt{4(k^2+1)n^2 + 12(k^2+1)n + (9k^2+2k+9)}\ \}.$$

Therefore, there are at most

$$\frac{1}{4}\{-2nk\ -\ 3k\ +\ 3\ +\ \sqrt{4(k^2+1)n^2 + 12(k^2+1)n + (9k^2+2k+9)}\}$$

terms of $g(j)$ which are nonnegative and

$$\sum_{j=(n-k)/2}^{n-k} \left\{ \frac{(n+1)(2j+k+1-n)(j+1) - (n+2)(2j+k-n)(k+j+1)}{k+j+1} \right\} \frac{\binom{n}{k+j}}{\binom{n}{j}}$$
$$\leq \frac{(n+1)(n+2-k)\{-2nk-3k+3+\sqrt{4(k^2+1)n^2+12(k^2+1)n+(9k^2+2k+9)}\}}{4(n+2+k)}.$$

To show that $W(n+1,k) - W(n,k)\ \geq\ 0$, now it is sufficient to show that

$$\frac{(n+1)(n+2-k)\{-2nk-3k+3+\sqrt{4(k^2+1)n^2+12(k^2+1)n+(9k^2+2k+9)}\}}{4(n+2+k)}$$
$$\leq \frac{(n+2+k)(n+2-k)}{4},$$

which is equivalent to showing that

$$(n+1)\{-2nk-3k+3+\sqrt{4(k^2+1)n^2+12(k^2+1)n+(9k^2+2k+9)}\} \leq (n+2+k)^2.$$

To show that

$$(n+1)\{-2nk-3k+3+\sqrt{4(k^2+1)n^2+12(k^2+1)n+(9k^2+2k+9)}\} \leq (n+2+k)^2,$$

it is sufficient to show that

$$(n+1)^2\{4(k^2+1)n^2+12(k^2+1)n+(9k^2+2k+9)\} \leq \{(n+1)(2nk+3k-3)+(n+2+k)^2\}^2.$$

After simplification,

$$\{(n+1)(2nk+3k-3)+(n+2+k)^2\}^2\ -\ (n+1)^2\{4(k^2+1)n^2+12(k^2+1)n+(9k^2+2k+9)\}$$
$$= (4k-3)n^4\ +\ (8k^2+18k-6)n^3\ +\ (4k^3+42k^2+30k-10)n^2\ +\ (14k^3+70k^2+24k-16)n$$
$$+\ (k^4+14k^3+42k^2+12k-8).$$

Since $1 \leq k \leq n$,

$$(4k-3)n^4\ +\ (8k^2+18k-6)n^3\ +\ (4k^3+42k^2+30k-10)n^2\ +\ (14k^3+70k^2+24k-16)n$$
$$+\ (k^4+14k^3+42k^2+12k-8) \geq 0.$$

This completes the proof of Theorem 4. □

THEOREM 5. $W(n, k_n)\ \approx\ \frac{n}{4}$.

Proof. Notice that for each fixed k,

$$\frac{1}{n^2}\sum_{j=(n-k)/2}^{n-k}(2j+k-n)\frac{\binom{n}{k+j}}{\binom{n}{j}} \approx \int_{1/2}^{1}(2x-1)(1-x)^k x^{-k}dx$$

and

$$\frac{1}{n^2}\sum_{j=(n-k+1)/2}^{n-k}(2j+k-n)\frac{\binom{n}{k+j}}{\binom{n}{j}} \approx \int_{1/2}^{1}(2x-1)(1-x)^k x^{-k}dx.$$

Let $t=(1-x)/x$; then

$$\int_{1/2}^{1}(2x-1)(1-x)^k x^{-k}dx = \int_0^1(1-t)t^k(1+t)^{-3}dt.$$

By the mean value theorem,

$$\int_0^1(1-t)t^k(1+t)^{-3}dt = a_k\int_0^1(1-t)t^k dt = \frac{a_k}{(k+1)(k+2)}$$

for some constant $\frac{1}{8}<a_k<1$. Therefore, $\frac{4W(n,k)}{n} \approx (1-\frac{k}{n})^2-\frac{b_k}{(k+1)(k+2)}$ as $n\to\infty$ for any fixed positive integer k, where b_k is a constant between $\frac{1}{2}$ and 4. Since k is arbitrary and $\frac{4W(n,k_n)}{n}\geq\frac{4W(n,k)}{n}$ for all $1\leq k\leq n$, $\frac{4W(n,k_n)}{n}\to 1$ as $n\to\infty$ and this completes the proof of Theorem 5. □

Combining Theorems 2 and 5, we have the following theorem.

THEOREM 6. *The k_n in the hole drawing policy is asymptotically optimal. Even though the computation for $W(n,k)$ is much simpler and faster than that for $E(n,0,0)$, we still have to compute $W(n,k)$ for all k, $1\leq k\leq n$ to identify k_n. The next result enables us to reduce the amount of required computation somewhat.*

THEOREM 7. *For $1\leq k\leq n-4$, if $W(n,k)\geq W(n,k+2)$, then $W(n,k)\geq W(n,k+2j)$ for all $1\leq j\leq(n-k)/2$.*

Proof. We will give the proof for the case when $n-k$ is even since the proof for the case when $n-k$ is odd is similar. By a direct computation, Theorem 7 holds for $1\leq n<10$, so we will assume that $n\geq 10$ in the proof below. It is also easy to verify that Theorem 7 holds if $k=n-4$; we will assume $n-k\geq 6$. For each $1\leq k\leq n$, let $u(n,k)=\frac{(n-k)(n-k+2)}{4}$ and $v(n,k)=\sum_{j=(n-k)/2}^{n-k}(2j+k-n)\frac{\binom{n}{k+j}}{\binom{n}{j}}$. Also for $1\leq k\leq n-2$, let $u'(n,k)=u(n,k)-u(n,k+2)$ and $v'(n,k)=v(n,k)-v(n,k+2)$. It is easy to see that $u'(n,k)=n-k$ for all $1\leq k\leq n-2$ and $n-k$ is even. Since $\frac{\binom{n}{j+1+k}}{\binom{n}{j+1}}\geq\frac{\binom{n}{j+k+2}}{\binom{n}{j}}$ if $j\geq\frac{n-k}{2}-1$, $v'(n,k)\geq 0$. It is also clear that $(n+1)W'(n,k)=(n+1)\{W(n,k)-W(n,k+2)\}=u'(n,k)-v'(n,k)$. Now we will prove that for fixed n, there exists a k'_n such that $v'(n,k)\geq u'(n,k)$ if $k\leq k'_n$ and $v'(n,k)<u'(n,k)$ if $k'_n<k<n-2$. Since $u'(n,k)$ is a linear function in k, it is sufficient to prove that $v'(n,k)$ is convex in k, i.e., to prove that $v'(n,k)+v'(n,k+4)\geq 2v'(n,k+2)$. It is equivalent to proving that $v(n,k)-3v(n,k+2)+3v(n,k+4)-v(n,k+6)\geq 0$. After

simplification,

$$v(n,k) - 3v(n,k+2) + 3v(n,k+4) - v(n,k+6)$$

$$= \sum_{(n-k)/2}^{n-k} (2j+k-n) \left\{ \frac{\binom{n}{j+k}}{\binom{n}{j}} - 3\frac{\binom{n}{j+k+1}}{\binom{n}{j-1}} + 3\frac{\binom{n}{j+k+2}}{\binom{n}{j-2}} - \frac{\binom{n}{j+k+3}}{\binom{n}{j-3}} \right\}.$$

It is easy to see that, to show $v(n,k)-3v(n,k+2)+3v(n,k+4)-v(n,k+6) \geq 0$, it is sufficient to show that for $n-k$ even, $n \geq 10,\ n-k \geq 6$, and $\frac{(n-k)}{2} \leq j \leq n-k-3$,

$$\frac{\binom{n}{j+k}}{\binom{n}{j}} - 3\frac{\binom{n}{j+k+1}}{\binom{n}{j-1}} + 3\frac{\binom{n}{j+k+2}}{\binom{n}{j-2}} - \frac{\binom{n}{j+k+3}}{\binom{n}{j-3}} \geq 0.$$

Since for $j = n-k-2,\ n-k-1,\ n-k$,

$$\frac{\binom{n}{j+k}}{\binom{n}{j}} - 3\frac{\binom{n}{j+k+1}}{\binom{n}{j-1}} + 3\frac{\binom{n}{j+k+2}}{\binom{n}{j-2}} - \frac{\binom{n}{j+k+3}}{\binom{n}{j-3}} \geq 0,$$

it is sufficient to show that for $n-k$ even, $n \geq 10,\ n-k \geq 6$, and $\frac{(n-k)}{2} \leq j \leq n-k-3$,

$$\frac{\binom{n}{j+k}}{\binom{n}{j}} - 3\frac{\binom{n}{j+k+1}}{\binom{n}{j-1}} + 3\frac{\binom{n}{j+k+2}}{\binom{n}{j-2}} - \frac{\binom{n}{j+k+3}}{\binom{n}{j-3}} = \frac{(j-3)!(n-j)!}{(j+k+3)!(n-j-k)!} h(j,k,n) \geq 0,$$

where

$$h(j,k,n) = j(j-1)(j-2)(j+k+1)(j+k+2)(j+k+3)$$

$$-3(j-1)(j-2)(n-j+1)(j+k+2)(j+k+3)(n-k-j)$$

$$+3(j-2)(n-j+1)(n-j+2)(j+k+3)(n-k-j)(n-k-j-1)$$

$$-(n-j+1)(n-j+2)(n-j+3)(n-k-j)(n-k-j-1)(n-k-j-2).$$

Let $n-k = 2y,\ j = y+x,\ k = z, n = 2y+z$; then

$$h(j,k,n) = h(y+x,z,2y+z) = 8(8x^3+x)y^3 + \{12(8x^3-4x^2+x)z + 12(8x^3-12x^2+x)\}y^2$$

$$+\{12(4x^3-4x^2+x)z^2 + 48(2x^3-4x^2+x)z + 4(14x^3-36x^2+13x)\}y$$

$$+\{4(2x^3-3x^2+x)z^3 + 12(2x^3-5x^2+2x)z^2 + 4(7x^3-21x^2+11x)z + 12(x^3-3x^2+24x)\}.$$

Since $x = 0,1,2,\ldots,y-3,\ y = 3,4,\ldots,\ z = 2,4,\ldots,$ it is easy to check that $h(y+x,z,2y+z) \geq 0$. The proof of Theorem 7 now is complete. □

We can use Theorem 7 to identify k_n by comparing $W(n,k)$ and $W(n,k+2)$. Once we find k_1 and k_2 such that $W(n,2k_1-1) \geq W(n,2k_1+1)$ and $W(n,2k_2) \geq W(n,2k_2+2)$, then $k_n = 2k_1-1$ if $W(n,2k_1-1) \geq W(n,2k_2)$ and $k_n = 2k_2$ if $W(n,2k_1-1) < W(n,2k_2)$. The problem here is that we still don't know how many values of $W(n,k)$ we will have to compute. Fortunately the next theorem gives a lower bound and an upper bound for k_n, which helps us to reduce the amount of required computation to identify k_n.

THEOREM 8. *Let k_n, $\underline{k}_n$, and $\overline{k}_n$ be as defined above. Then for all $n \geq 1$, $\underline{k}_n \leq k_n \leq \overline{k}_n$.*

Proof. We will give the proof for $k_n \leq \overline{k}_n$ since the proof for $\underline{k}_n \leq k_n$ is similar. We also give the proof only for the case when $n-k$ is even since the proof for the case when $n-k$ is odd is also similar. We will assume that $n \geq 1{,}000$ since Table 2 above reveals that Theorem 8 holds for $n \leq 1{,}000$. By Theorem 7, if $v'(n,k) < u'(n,k) = n-k$ and $v'(n,k-1) < u'(n,k-1) = n-k+1$, then $k_n \leq k$. Now

$$v'(n,k) = v(n,k) - v(n,k+2) = \frac{(n-k)}{\binom{n}{k}} + \sum_{j=k+1}^{(n+k)/2} (n+k-2j) \left\{ \frac{\binom{n}{j-k}}{\binom{n}{j}} - \frac{\binom{n}{j-k-1}}{\binom{n}{j+1}} \right\}$$

$$= \frac{(n-k)}{\binom{n}{k}} + (n+1) \sum_{j=k+1}^{(n+k)/2} (n+k-2j)^2 \frac{j!\,(n-j-1)!}{(j-k)!\,(n-j+k+1)!}$$

$$< \frac{(n-k)}{\binom{n}{k}} + (n+1) \sum_{j=k+1}^{(n+k)/2} \frac{(n+k-2j)^2}{(n-j)(n+k+1-j)} \left(\frac{j}{n+k-j}\right)^k$$

$$< \frac{(n-k)}{\binom{n}{k}} + \frac{4(n+1)}{(n-k)(n+k)} \sum_{j=k+1}^{(n+k)/2} (n+k-2j)^2 \left(\frac{j}{n+k-j}\right)^k$$

$$\approx \frac{(n-k)}{\binom{n}{k}} + \frac{4(n+1)(n+k)^2}{(n-k)} \int_{1/2}^{1} (2x-1)^2(1-x)^k x^{-k} dx.$$

For $2 \leq k \leq n-2$ and n large, $\frac{(n-k)}{\binom{n}{k}}$ is negligible. Also notice that

$$\int_{1/2}^{1} (2x-1)^2(1-x)^k x^{-k} dx = \lim_{m\to\infty} \sum_{j=1}^{2mk} \frac{1}{4mk} \frac{j^2}{4m^2k^2} \left(\frac{2mk-j}{2mk+j}\right)^k$$

$$\leq \lim_{m\to\infty} \sum_{j=1}^{2mk} \frac{1}{16k^3} \frac{j^2}{m^3} e^{-j/m} \leq \lim_{m\to\infty} \frac{1}{16k^3} \frac{1}{m^3} \frac{(e^{2/m}+e^{1/m})}{(e^{1/m}-1)^3} = \frac{1}{8k^3}.$$

Therefore, $v'(n,k) = v(n,k) - v(n,k+2) < \frac{(n+1)(n+k)^2}{2(n-k)k^3}$. Now if $(n+1)(n+k)^2 \leq 2(n-k)^2k^3$, then $v'(n,k) < \frac{(n+1)(n+k)^2}{2(n-k)k^3} \leq (n-k)$ and $k_n \leq k$ and this completes the proof of Theorem 8. □

Table 3 provides some numerical values of $\underline{k}_n, k_n, k_n^*, \overline{k}_n, W(n,k_n)$, and $\frac{(n-k_n+1)^2}{4(n+1)}$ for $n = 100, 200, 300, \ldots, 3{,}000$, where $k_n^* =$ the integer part of $(\frac{n}{2})^{1/3}$.

By Theorems 7 and 8, we can identify the optimal k_n very quickly. From the proof of Theorem 5, $W(n,k) = \frac{1}{n+1}\{\frac{(n-k+2)(n-k)}{4} - \frac{n^2c_k}{(k+1)(k+2)}\}$ if $n-k$ is even and $W(n,k) = \frac{1}{n+1}\{\frac{(n-k+1)^2}{4} - \frac{n^2c_k}{(k+1)(k+2)}\}$ if $n-k$ is odd, where c_k is a constant between $\frac{1}{2}$ and 1. Therefore, the optimal $k_n = d_n n^{1/3}$, where d_n is a constant less than 1.

Table 3
$\underline{k}_n, k_n, k_n^*, \overline{k}_n, W(n, k_n)$ and $\frac{(n-k_n+1)^2}{4(n+1)}$.

n	$\underline{k}_n$	k_n	k_n^*	$\overline{k}_n$	$W(n,k_n)$	$\frac{(n-k_n+1)^2}{4(n+1)}$
100	2	3	3	4	22.59	23.77
200	2	4	4	5	46.86	48.27
300	3	5	5	6	71.37	72.77
400	3	6	6	6	95.95	97.27
500	3	6	6	7	120.62	122.27
600	4	7	7	7	145.30	146.77
700	4	7	7	8	170.05	171.77
800	4	7	7	8	194.80	196.77
900	4	8	8	8	219.56	221.27
1,000	4	8	8	9	244.37	246.27
1,100	5	8	8	9	269.18	271.26
1,200	5	8	8	9	293.99	296.26
1,300	5	9	8	9	318.81	320.77
1,400	5	9	8	9	343.65	345.76
1,500	5	9	9	10	368.50	370.76
1,600	5	9	9	10	393.35	395.76
1,700	5	9	9	10	418.20	420.76
1,800	6	10	9	10	443.06	445.26
1,900	6	10	9	10	467.93	470.26
2,000	6	10	10	11	492.81	495.26
2,100	6	10	10	11	517.69	520.26
2,200	6	10	10	11	542.56	545.26
2,300	6	10	10	11	567.44	570.26
2,400	6	11	10	11	592.33	594.76
2,500	6	11	10	11	617.22	619.76
2,600	6	11	10	11	642.12	644.76
2,700	6	11	11	12	667.02	669.76
2,800	7	11	11	12	691.91	694.76
2,900	7	11	11	12	716.81	719.76
3,000	7	11	11	12	741.71	744.76

By Theorem 8, the constant d_n is approximately between $\frac{1}{2}$ and $(\frac{1}{2})^{1/3}$. From Table 3, it seems that $k_n =$ the integer part of $\{\frac{1}{2} + (\frac{n}{2})^{1/3}\}$. However, we do not have a proof yet. We can start with $k =$ the integer part of $\{\frac{1}{2} + (\frac{n}{2})^{1/3}\}$ and compare $W(n,k), W(n,k+2)$ and $W(n,k+1), W(n,k+3)$. Then we either increase k by 1 or decrease k by 1. By this procedure, we can identify k_n very quickly. For example, even when $n =$ 100,000, we need at most 14 comparisons to identify the optimal k_n. Table 3 also confirms Theorem 8, which implies that $k_n \to \infty$ as $n \to \infty$ even though $k_n \to \infty$ very slowly.

Theorem 9. *$k_n \to \infty$ as $n \to \infty$.*

Although we are not able to prove that k_n is nondecreasing in n, we have the following weaker theorem, which is interesting and useful to identify k_n.

Theorem 10. *For all $n \geq 1$, $|k_{n+1} - k_n| \leq 1$.*

Proof. By Theorem 7, it is sufficient to prove $v(n+1,k-1) - v(n+1,k) \geq v(n,k) - v(n,k+1) \geq v(n+1,k+1) - v(n+1,k+2)$. We will give only the proof for $v(n,k) - v(n,k+1) \geq v(n+1,k+1) - v(n+1,k+2)$ since the proof for $v(n+1,k-1) - v(n+1,k) \geq v(n,k) - v(n,k+1)$ is similar. We will assume that

$n+k$ is even since the proof for the case when $n+k$ is odd is similar. Notice that

$$\begin{aligned} &v(n,k) - v(n,k+1) \\ &\quad = \sum_{j=k}^{(n+k)/2-1} (n+k-2j) \frac{\binom{n}{j-k}}{\binom{n}{j}} - \sum_{j=k}^{(n+k)/2-1} (n+k-1-2j) \frac{\binom{n}{j-k}}{\binom{n}{j+1}} \\ &\quad = \sum_{j=k}^{(n+k)/2-1} \frac{\binom{n}{j-k}}{\binom{n}{j}} \left\{ (n+k-2j) - (n+k-1-2j) \frac{j+1}{n-j} \right\} \end{aligned}$$

and

$$\begin{aligned} &v(n+1,k+1) - v(n+1,k+2) \\ &\quad = \sum_{j=k+1}^{(n+k)/2} (n+k+2-2j) \frac{\binom{n+1}{j-k-1}}{\binom{n+1}{j}} - \sum_{j=k+1}^{(n+k)/2} (n+k+1-2j) \frac{\binom{n+1}{j-k-1}}{\binom{n+1}{j+1}} \\ &\quad = \sum_{j=k}^{(n+k)/2-1} (n+k-2j) \frac{\binom{n+1}{j-k}}{\binom{n+1}{j+1}} - \sum_{j=k}^{(n+k)/2-1} (n+k-1-2j) \frac{\binom{n+1}{j-k}}{\binom{n+1}{j+2}} \\ &\quad = \sum_{j=k}^{(n+k)/2-1} \frac{\binom{n+1}{j-k}}{\binom{n+1}{j+1}} \left\{ (n+k-2j) - (n+k-1-2j) \frac{j+2}{n-j} \right\} \\ &\quad = \sum_{j=k}^{(n+k)/2-1} \frac{\binom{n}{j-k}}{\binom{n}{j}} \left\{ (n+k-2j) - (n+k-1-2j) \frac{j+2}{n-j} \right\} \frac{j+1}{n+k+1-j}. \end{aligned}$$

Since $1 \le j \le \frac{n+k}{2} - 1$, $\frac{j+1}{n+k+1-j} < 1$. Therefore, $v(n,k) - v(n,k+1) \ge v(n+1,k+1) - v(n+1,k+2)$ and this completes the proof of Theorem 10. $\square$

From our computation, we notice that k_n is nondecreasing in n. If this statement is true, we can further reduce the computation for identifying k_n. However, we do not have a proof for this statement either. It is worthwhile to point out that both "the k_n^* in the hole drawing policy" and "the $\overline{k}_n$ in the hole drawing policy" are also asymptotically optimal. However, we do not know how big the difference between $W(n, k_n)$ and $W(n, k_n^*)$ will be. If the difference between $W(n, k_n)$ and $W(n, k_n^*)$ is bounded, then we can just use "the k_n^* in the hole drawing policy."

4. A random coin tossing problem. In this section, we will show that our urn problem is in fact equivalent to the following coin tossing problem, which can be described as follows: A player is given a coin and is allowed to toss the coin at most n times, but can stop any time he or she wishes. The player gets a $+1$ each time a head is tossed and a -1 each time a tail is tossed. The player does not know the probability "p" of getting a head on each toss but knows that p has a uniform distribution over the interval $[0,1]$.

For each positive integer n and integers $0 \leq j \leq k \leq n$, let $G(n,k,j)$ be the player's additional (conditional) expected value at the stopping time when he or she uses an optimal stopping rule for the remaining game given that the player has tossed the coin k times and j of which are heads.

LEMMA 2. *If $0 \leq j \leq k \leq n-1$, then*

$$G(n,k,j) = \max\left\{0, \frac{2j-k}{k+2} + \frac{j+1}{k+2}G(n,k+1,j+1) + \frac{k-j+1}{k+2}G(n,k+1,j)\right\}.$$

By mathematical induction, it is easy to show that for fixed n and k, $G(n,k,j)$ is increasing in j. It makes sense to define j_{nk} to be the smallest j such that $G(n,k,j) > 0$. The optimal stopping rule can then be stated as follows: If the player did not stop earlier and has tossed the coin k times, j of which are heads, then the player should continue to toss as long as $j \geq j_{nk}$ unless $k = n$. It is clear that $E(n,k,j) = G(n,k,j)$ for all $0 \leq j \leq k \leq n$ and $n \geq 1$, so this random coin tossing problem is equivalent to the random version of Shepp's urn scheme problem.

For each nonnegative integer k, "the k in the hole stopping rule" says the player will continue to toss the coin if the number of tails tossed is still less than k + the number of heads tossed. Let $H(n,k)$ be the expected value of the game when the player uses "the k in the hole stopping rule."

THEOREM 11. *For all $1 \leq k \leq n$, $H(n,k) = W(n,k)$.*

Proof. It is sufficient to show that if $n-k$ is even,

$$H(n,k) = \frac{1}{n+1}\left\{\frac{(n-k+2)(n-k)}{4} - \sum_{j=(n-k)/2}^{n-k}(2j+k-n)\frac{\binom{n}{k+j}}{\binom{n}{j}}\right\},$$

and if $n-k$ is odd,

$$H(n,k) = \frac{1}{n+1}\left\{\frac{(n-k+1)^2}{4} - \sum_{j=(n-k+1)/2}^{n-k}(2j+k-n)\frac{\binom{n}{k+j}}{\binom{n}{j}}\right\}.$$

We give the proof for the case when $n-k$ is even; when $n-k$ is odd the proof is similar. For each $j = 0,1,2,\ldots,n$, let $H(n,k,j)$ be the value at the stopping time when the player uses "the k in the hole stopping rule" assuming that there are j heads in n tosses. Then it is clear that $H(n,k,j) = 2j-n$ if $j > n-k$,

$$H(n,k,j) = (2j-n)\left\{1 - \binom{n}{k+j}\bigg/\binom{n}{k}\right\} - k\binom{n}{k+j}\bigg/\binom{n}{k}$$

if $(n-k)/2 \leq j \leq n-k$ (by the reflection principle), and $H(n,k,j) = -k$ if $j < (n-k)/2$. For each $j = 0,1,2,\ldots,n$, let $P(j)$ be the probability of getting j heads in n tosses. Given that the probability of getting a head in a toss is p, $P(j) = \binom{n}{j}p^j(1-p)^{n-j}$. Hence

$$\begin{aligned} H(n,k) &= \int_0^1 \sum_{j=0}^{n} H(n,k,j)P(j)dp \\ &= \frac{1}{n+1}\left\{\frac{(n-k+2)(n-k)}{4} - \sum_{j=(n-k)/2}^{n-k}(2j+k-n)\frac{\binom{n}{k+j}}{\binom{n}{j}}\right\}. \end{aligned}$$

Therefore, $H(n,k) = W(n,k)$. □

Acknowledgments. We would like to sincerely thank the referee for invaluable suggestions and comments which improved the presentation of this paper significantly. We also want to thank Professor Donald Berry for mentioning this Bayesian (random) version of Shepp's urn scheme to us. R. Chen would like to express his sincere thanks to the Department of Mathematics of Tamkang University of Taiwan, especially to Professor S. J. Hu.

REFERENCES

[1] W. M. Boyce, *Stopping rules for selling bonds*, RAND J. Econ., 1 (1970), pp. 27–53.

[2] W. M. Boyce, *On a simple optimal stopping problem*, Discrete Math., 5 (1973), pp. 297–312.

[3] R. W. Chen and F. K. Hwang, *On the values of an (m,p) urn*, Congr. Numer., 41 (1984), pp. 75–84.

[4] R. W. Chen, A. M. Odlyzko, L. A. Shepp, and A. Zame, *An optimal acceptance policy for an urn scheme*, SIAM J. Discrete Math., 11 (1998), pp. 183–195.

[5] W. Feller, *An Introduction to Probability Theory and Its Applications*, 3rd Edition, John Wiley & Sons, New York, 1967.

[6] H. W. Gould, *Combinatorial Identities: A Standardized Set of Tables Listing* 500 *Binomial Coefficient Summations*, Henry W. Gould, Morgantown, WV., 1972.

[7] N. L. Johnson and S. Kotz, *Urn Models and Their Applications*, John Wiley & Sons, New York, 1977.

[8] L. A. Shepp, *Explicit solutions to some problems of optimal stopping*, Ann. Math. Statist., 40 (1969), pp. 993–1010.

SIAM J. DISCRETE MATH.
Vol. 19, No. 1, pp. 165–191

ALTERNATIVE DIGIT SETS FOR NONADJACENT REPRESENTATIONS[*]

JAMES A. MUIR[†] AND DOUGLAS R. STINSON[‡]

Abstract. It is known that every positive integer n can be represented as a finite sum of the form $n = \sum a_i 2^i$, where $a_i \in \{0, 1, -1\}$ for all i, and no two consecutive a_i's are nonzero. Such sums are called *nonadjacent representations*. Nonadjacent representations are useful in efficiently implementing elliptic curve arithmetic for cryptographic applications.

In this paper, we investigate if other digit sets of the form $\{0, 1, x\}$, where x is an integer, provide each positive integer with a nonadjacent representation. If a digit set has this property, we call it a *nonadjacent digit set* (NADS). We present an algorithm to determine if $\{0, 1, x\}$ is an NADS and, if it is, we present an algorithm to efficiently determine the nonadjacent representation of any positive integer. We also present some necessary and sufficient conditions for $\{0, 1, x\}$ to be an NADS. These conditions are used to exhibit infinite families of integers x such that $\{0, 1, x\}$ is an NADS, as well as infinite families of x such that $\{0, 1, x\}$ is not an NADS.

Key words. nonadjacent representations, redundant representations, digit sets, radix 2

AMS subject classifications. 11A63, 14G50

DOI. 10.1137/S0895480103437651

1. Introduction and history.

In a base 2 (or *radix* 2) positional number system, representations of integers are converted into integers via the rule

$$(\dots a_3 a_2 a_1 a_0)_2 = \cdots + a_3 2^3 + a_2 2^2 + a_1 2^1 + a_0.$$

Each of the a_i's is called a *digit*. In the usual radix 2 positional number system each digit is equal to 0 or 1. If we let $D = \{0, 1\}$, then we say that D is the *digit set* for this number system.

It is often advantageous to employ alternate digit sets. The digit set $D = \{0, 1, -1\}$ was studied as early as 1951 by Booth. Booth [1] presents a technique whereby a binary computer can calculate a representation of the product of two integers without any extra steps to correct for its sign. His method is implicitly based on replacing one of the operands in the multiplication with a $\{0, 1, -1\}$-radix 2 representation. In 1960, through his investigations on how to reduce the number of additions and subtractions used in binary multiplication and division, Reitwiesner [8] showed that every integer has an easily constructed canonical $\{0, 1, -1\}$-radix 2 representation with a *minimal* number of nonzero digits.

Reitwiesner's canonical $\{0, 1, -1\}$-radix 2 representations are defined by the following property.

[*] Received by the editors November 18, 2003; accepted for publication (in revised form) November 21, 2004; published electronically June 22, 2005. This is an extended version of the paper which appeared in *Selected Areas in Cryptography. Revised Papers from the* 10th *Annual International Workshop* (*SAC* 2003)*, Ottawa, ON, August* 14–15, 2003, M. Matsui and R. Zuccherato, eds., Lecture Notes in Comput. Sci. 3006, Springer-Verlag, Berlin, 2004, pp. 306–319.

http://www.siam.org/journals/sidma/19-1/43765.html

[†] Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada (jamuir@uwaterloo.ca, http://www.math.uwaterloo.ca/∼jamuir). This author's research was supported in part by an NSERC (Canada) Postgraduate Scholarship.

[‡] School of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada (dstinson@uwaterloo.ca, http://www.cacr.math.uwaterloo.ca/∼dstinson). This author's research was supported by NSERC (Canada) grant RGPIN 203114-02.

Property M. *Of any two consecutive digits, at most one is nonzero.*

In other words, in such representations, nonzero digits are nonadjacent. These representations have come to be called *nonadjacent forms* (NAFs). The terminology "Property M" was applied by Reitwiesner and likely reflects the fact that $\{0, 1, -1\}$-radix 2 representations, which satisfy this property, have a minimal number of nonzero digits.

Cryptographers came to be interested in NAFs through the study of exponentiation. Jedwab and Mitchell [4] noticed that it is possible to reduce the number of multiplications used in the square-and-multiply algorithm for exponentiation if a $\{0, 1, -1\}$-radix 2 representation of the exponent is used. This led them to an independent discovery of the NAF. However, in multiplicative groups, like those used in the Rivest–Shamir–Adleman (RSA) public key cryptosystem and the Digital Signature Algorithm (DSA) [6], using the digit -1 requires the computation of an inverse, which is more costly than a multiplication.

In elliptic curve groups this is not a problem since inverses can be computed essentially for free. Morain and Olivos [7] observed that in these groups the operation analogous to exponentiation could be made more efficient using $\{0, 1, -1\}$ representations. They give two algorithms for performing scalar-multiplication using addition and subtraction. The $\{0, 1, -1\}$-radix 2 representations, upon which their algorithms are based, are in fact the same ones that Booth and Reitwiesner studied. NAFs and representations like them are important tools in the efficient implementations of elliptic curve cryptosystems (cf. Gordon [3], Solinas [10, 11], Brickell et al. [2]).

If a radix 2 representation has digit set D and satisfies Property M, we call it a *D-nonadjacent form* (D-NAF). In this paper, we consider the question of which sets D provide NAFs for *every positive* integer. If D is such a digit set, then we call it a *nonadjacent digit set* (NADS). After a preliminary version of this paper was completed, it was discovered that a related question had been studied by Matula. Matula [5] defines and investigates *basic* digit sets. A set of digits containing 0 is called *basic* if it provides every positive and negative integer with a unique radix-r representation without the use of a separate sign. If a digit set is basic, Matula shows that $r \neq 2$; in this paper we are concerned only with radix 2 representations. Another difference between our work and Matula's is that he imposes no relation on the digits of a representation, while we are interested only in nonadjacent representations.

We examine digit sets of the form $\{0, 1, x\}$ with $x \in \mathbb{Z}$. It is known that letting $x = -1$ gives an NADS, but it is somewhat surprising that there are many values of x with this property; for example, $x = -5, -13, -1145$. We give infinite families of x's for which $\{0, 1, x\}$ is an NADS, and we also give infinite families of x's for which $\{0, 1, x\}$ is not an NADS. We also give some results on the necessary conditions D must satisfy in order to be an NADS. The algorithms we present and analyze for computing D-NAFs might be of some interest as well.

2. Preliminaries.

We start by introducing some definitions and notation, which will facilitate our discussions.

All of the radix 2 representations we consider here are finite sums of the form $\sum_{i \geq 0} a_i 2^i$, which we denote by $(\ldots a_2 a_1 a_0)_2$. Since $(\ldots a_2 a_1 a_0)_2$ stands for a finite sum, all but a finite number of the a_i's are zero. Because of this property, we can consider the *length* of a representation as follows.

Definition 2.1. *The* length *of a representation* $(\ldots a_2 a_1 a_0)_2$ *is the integer*

$$\min\{\ell \in \mathbb{Z} : \ell \geq 0, \ \textit{and for any } i \geq \ell, \ a_i = 0\}.$$

For the representation $(a_{\ell-1}\dots a_1a_0)_2$, it is implicit that $a_i = 0$ for all $i \geq \ell$; note that if $a_{\ell-1} \neq 0$, then this representation has length ℓ. According to the definition above, the all-zero representation has length 0.

We will always use D to denote a digit set. The set of all *strings* of digits from D is denoted by D^*. The empty string is in D^* and is denoted by ϵ. Now, if D is the digit set for $(a_{\ell-1}\dots a_1a_0)_2$, then $a_{\ell-1}\dots a_1a_0$ is a string in D^*. Conversely, any string $\alpha \in D^*$ corresponds to a radix 2 representation with digit set D, namely, $(\alpha)_2$. If $\alpha, \beta \in D^*$, then we denote their *concatenation* by $\alpha\|\beta$.

We apply some of our terminology for representations to strings. If $0 \in D$, and if a finite string $\alpha \in D^*$ satisfies Property M, then we call α a D-NAF. If, in addition, $(\alpha)_2 = n$, we say α is a D-NAF for n. Notice that if α is a D-NAF for n, then α with any leading zeros removed is also a D-NAF for n. We denote the string formed by deleting the leading zeros from α by $\widehat{\alpha}$.

Given a digit set D and an integer n, we define a map

$$R_D(n) := \begin{cases} \widehat{\alpha}, & \text{where } \alpha \in D^* \text{ is a } D\text{-NAF for } n, \text{ if one exists,} \\ \perp & \text{otherwise.} \end{cases}$$

Here, $\perp$ is just some symbol not in D. If $R_D(n)$ evaluates to a D-NAF for n, then by definition that string has no leading zeros. For example, if $D = \{0, 1, -9\}$, then $R_D(7)$ might evaluate to $1000\overline{9}$ since $1000\overline{9}$ is a D-NAF, has no leading zeros, and $(1000\overline{9})_2 = 1\cdot 2^4 + 0\cdot 2^3 + 0\cdot 2^2 + 0\cdot 2^1 - 9\cdot 2^0 = 7$. If there is more than one string in D, which is a D-NAF for n and has no leading zeros, then $R_D(n)$ might evaluate to any one of these strings. Later on we will prove that 3 does not have a D-NAF, and hence $R_D(3) = \perp$.

We are interested in determining which integers have D-NAFs, so we define the set

$$\mathrm{NAF}(D) := \{n \in \mathbb{Z} : R_D(n) \neq \perp\}.$$

From our example with $D = \{0, 1, -9\}$ we see $7 \in \mathrm{NAF}(D)$, but $3 \notin \mathrm{NAF}(D)$. Using this notation, our definition of an NADS is as follows.

DEFINITION 2.2. *D is an* NADS *if* $\mathbb{Z}^+ \subseteq \mathrm{NAF}(D)$.

3. Necessary conditions for $\{0, 1, x\}$ to be an NADS.

If we suppose $D = \{0, 1, x\}$ is an NADS, then we can deduce necessary conditions on x.

THEOREM 3.1. *Let $D = \{0, 1, x\}$. If there exists $n \in \mathrm{NAF}(D)$ with $n \equiv 3 \pmod 4$, then $x \equiv 3 \pmod 4$.*

Proof. Take $n \in \mathrm{NAF}(D)$ with $n \equiv 3 \pmod 4$. For some particular D-NAF, say $(\dots a_2a_1a_0)_2$, we have

$$\begin{aligned} &(\dots a_2a_1a_0)_2 = n \\ &\qquad \Longrightarrow a_0 \equiv 1 \pmod 2 \\ &\qquad \Longrightarrow a_0 \neq 0. \end{aligned}$$

Since a_0 is nonzero and the representation is nonadjacent, we have $a_1 = 0$. Thus

$$\begin{aligned} &(\dots a_2 0 a_0)_2 = n \\ &\qquad \Longrightarrow a_0 \equiv 3 \pmod 4 \\ &\qquad \Longrightarrow a_0 \neq 1 \\ &\qquad \Longrightarrow a_0 = x. \end{aligned}$$

So $x = a_0 \equiv 3 \pmod 4$. □

If $D = \{0, 1, x\}$ is an NADS, then $3 \in \mathrm{NAF}(D)$, and by the previous result $x \equiv 3 \pmod 4$. So, if we are trying to find a value of x that makes $\{0, 1, x\}$ an NADS, we need only consider those values congruent to 3 modulo 4.

3.1. The case $x > 0$. If we restrict x to be a positive integer, then we can give a complete characterization of all values which make $D = \{0, 1, x\}$ an NADS. It is well known that $x = 3$ is such a value, and this is remarked by Solinas [9]. We give a proof of this fact and then show that no other positive value of x makes $\{0, 1, x\}$ an NADS.

Theorem 3.2. *The only NADS of the form* $\{0, 1, x\}$ *with* $x > 0$ *is* $\{0, 1, 3\}$.

Proof. Let n be any positive integer. We want to show that n has a $\{0, 1, 3\}$-NAF. Let $(\ldots a_2 a_1 a_0)_2$ be the usual $\{0, 1\}$-radix 2 representation of n. If this representation satisfies Property M, there is nothing to prove, so suppose it does not. Let i be the smallest integer for which $a_{i+1} = a_i = 1$. Replace digits a_{i+1} and a_i with 0 and 3, respectively. Since $2^{i+1} + 2^i = 0 \cdot 2^{i+1} + 3 \cdot 2^i$, the resulting representation stands for the same integer. By working from right to left, repeating this substitution as necessary, we transform $(\ldots a_2 a_1 a_0)_2$ into a $\{0, 1, 3\}$-NAF. This proves that $\{0, 1, 3\}$ is an NADS.

Now consider x with $x > 3$. We show $n = 3$ does not have a $\{0, 1, x\}$-NAF. Suppose to the contrary that for some $\{0, 1, x\}$-NAF we have $(\ldots a_2 a_1 a_0)_2 = 3$. Since 3 is odd, $a_0 \neq 0$ and so $a_1 = 0$. Now $a_0 \equiv 3 \pmod 4$, so it must be that $a_0 = x$. However, since each of the digits in $\{0, 1, x\}$ is nonnegative, we have

$$3 = (\ldots a_2 0 x)_2 = \cdots + a_2 2^2 + 0 \cdot 2^1 + x \geq x > 3,$$

which is a contradiction. So, 3 does not have a $\{0, 1, x\}$-NAF when $x > 3$. □

An example helps illustrate the construction used in the above proof. Suppose $n = 237$. To find a $\{0, 1, 3\}$-NAF for 237 we start with its usual binary representation and then, working from right to left, replace any occurrences of the digits 11 with 03:

$$237 = (11101101)_2 = (10300301)_2.$$

A natural question to ask is if this is the only $\{0, 1, 3\}$-NAF for 237. We give the answer in the next section.

3.2. Uniqueness. We show that every integer, not only just the positive ones, has at most one $\{0, 1, x\}$-NAF, where $x \equiv 3 \pmod 4$.

Theorem 3.3. *If* $x \equiv 3 \pmod 4$*, then any integer has at most one finite length* $\{0, 1, x\}$*-NAF.*

Proof. Let $D = \{0, 1, x\}$ and suppose the result is false. Then it must be that

$$(a_{\ell-1} \ldots a_2 a_1 a_0)_2 = (b_{\ell'-1} \ldots b_2 b_1 b_0)_2,$$

where $(a_{\ell-1} \ldots a_2 a_1 a_0)_2$ and $(b_{\ell'-1} \ldots b_2 b_1 b_0)_2$ are two different D-NAFs with lengths ℓ and ℓ', respectively. These representations stand for the same integer; call it n. We can assume that ℓ is as small as possible.

If $a_0 = b_0$, then

$$(a_{\ell-1} \ldots a_2 a_1)_2 = (b_{\ell'-1} \ldots b_2 b_1)_2,$$

and so we have two different, and shorter, D-NAFs, which stand for the same integer, contrary to the minimality of ℓ. So it must be that $a_0 \neq b_0$.

If one of a_0 or b_0 is 0, then n is even, and so both a_0 and b_0 are 0. But a_0 and b_0 are different so it must be that a_0 is equal to 1 or x. Without loss of generality, we can assume the representations have the form

$$(a_{\ell-1} \ldots a_2 0x)_2 = (b_{\ell'-1} \ldots b_2 01)_2.$$

This implies $x \equiv 1 \pmod 4$, contrary to our hypothesis that $x \equiv 3 \pmod 4$. Thus every integer has at most one D-NAF. □

4. Recognizing NADS of the form $\{0, 1, x\}$. From now on we fix $D = \{0, 1, x\}$ with $x \equiv 3 \pmod 4$. In this section we work toward a method of deciding if $\{0, 1, x\}$ is an NADS. By Theorem 3.2, this is easy when $x > 0$, so we will assume $x < 0$.

Recall that $R_D(n)$ evaluates either to the symbol $\perp$ or to a finite string, with no leading zeros, that is a D-NAF for n. Theorem 3.3 tells us that any n has at most one D-NAF, so in the second case, the string returned by $R_D(n)$ is unique. Thus, $R_D(n)$ is well defined (i.e., for every input n there is exactly one output).

The ability to evaluate $R_D(n)$ can be useful in deciding if D is an NADS. If we can find $n \in \mathbb{Z}^+$ such that $R_D(n) = \perp$, then we know that D is not an NADS. Also, if we have an algorithmic description of $R_D(n)$, we might be able to analyze this algorithm and show that for any $n \in \mathbb{Z}^+$, $R_D(n) \neq \perp$, thus proving that D is an NADS.

We show that $R_D(n)$ can be computed recursively and give an algorithm which evaluates $R_D(n)$ in this manner. We begin with some lemmas.

Lemma 4.1. *If $n \equiv 0 \pmod 4$, then $n \in \mathrm{NAF}(D)$ if and only if $n/4 \in \mathrm{NAF}(D)$. Further, if $n \in \mathrm{NAF}(D)$, then $R_D(n) = R_D(n/4)\|00$.*

Proof. Since $n \equiv 0 \pmod 4$, the definition of the digit set D implies that any D-NAF for n is of the form $(a_{\ell-1} \ldots a_3 a_2 00)_2$, where $a_{\ell-1} \neq 0$. Now,

$$\begin{aligned} n \in \mathrm{NAF}(D) &\iff n \text{ has a } D\text{-NAF of the form } (a_{\ell-1} \ldots a_3 a_2 00)_2 \\ &\iff n/4 \text{ has a } D\text{-NAF of the form } (a_{\ell-1} \ldots a_3 a_2)_2 \\ &\iff n/4 \in \mathrm{NAF}(D), \end{aligned}$$

which proves the first part of the lemma. If $n \in \mathrm{NAF}(D)$, then

$$R_D(n) = a_{\ell-1} \ldots a_3 a_2 00 = a_{\ell-1} \ldots a_3 a_2 \| 00 = R_D(n/4)\|00,$$

which proves the second part of the lemma. □

We omit the proofs of the next three lemmas since they can be established by making only minor changes to the proof of Lemma 4.1.

Lemma 4.2. *If $n \equiv 1 \pmod 4$, then $n \in \mathrm{NAF}(D)$ if and only if $(n-1)/4 \in \mathrm{NAF}(D)$. Further, if $n \in \mathrm{NAF}(D)$, then $R_D(n) = R_D(\frac{n-1}{4})\|01$.*

Lemma 4.3. *If $n \equiv 2 \pmod 4$, then $n \in \mathrm{NAF}(D)$ if and only if $n/2 \in \mathrm{NAF}(D)$. Further, if $n \in \mathrm{NAF}(D)$, then $R_D(n) = R_D(n/2)\|0$.*

Lemma 4.4. *If $n \equiv 3 \pmod 4$, then $n \in \mathrm{NAF}(D)$ if and only if $(n-x)/4 \in \mathrm{NAF}(D)$. Further, if $n \in \mathrm{NAF}(D)$, then $R_D(n) = R_D(\frac{n-x}{4})\|0x$.*

Given an integer n, if we somehow know that $n \in \mathrm{NAF}(D)$, then Lemmas 4.1–4.4 suggest a recursive procedure that we can use to evaluate $R_D(n)$. To illustrate, suppose $D = \{0, 1, -9\}$. It was shown in an earlier example that $7 \in \mathrm{NAF}(D)$. Using these lemmas, we have

$$R_D(7) = R_D(4)\|0\overline{9} = R_D(1)\|00\|0\overline{9} = 1\|00\|0\overline{9} = 1000\overline{9}.$$

To describe the general procedure for computing $R_D(n)$, given that $n \in \mathrm{NAF}(D)$, we use the following two functions:

$$f_D(n) := \begin{cases} n/4 & \text{if } n \equiv 0 \pmod 4, \\ (n-1)/4 & \text{if } n \equiv 1 \pmod 4, \\ n/2 & \text{if } n \equiv 2 \pmod 4, \\ (n-x)/4 & \text{if } n \equiv 3 \pmod 4, \end{cases} \tag{4.1}$$

$$g_D(n) := \begin{cases} 00 & \text{if } n \equiv 0 \pmod 4, \\ 01 & \text{if } n \equiv 1 \pmod 4, \\ 0 & \text{if } n \equiv 2 \pmod 4, \\ 0x & \text{if } n \equiv 3 \pmod 4. \end{cases} \tag{4.2}$$

Note that f_D returns an integer, and g_D returns a string. Below the procedure is described in pseudocode.

```
PROCEDURE 4.5. EVAL_α-$R_D(n)$

$\alpha \leftarrow \epsilon$
while $n \neq 0$
    do { $\alpha \leftarrow g_D(n) \parallel \alpha$
         $n \leftarrow f_D(n)$
return $\widehat{\alpha}$
```

Procedure 4.5 terminates on input n if and only if ${f_D}^i(n) = 0$ for some positive integer i. An easy calculation shows that, for $D = \{0, 1, -9\}$, ${f_D}^3(7) = 0$, and so the procedure terminates on input $n = 7$. However, $f_D(3) = 3$ and so ${f_D}^i(3) = 3 \neq 0$ for all i, and thus the procedure does not terminate on input $n = 3$.

Using the previous lemmas, we can show that Procedure 4.5 terminates on input n if and only if $n \in \mathrm{NAF}(D)$. Instead of making use of the lemmas individually, we find it more convenient to summarize them as follows.

LEMMA 4.6. *For all $n \in \mathbb{Z}$, $n \in \mathrm{NAF}(D)$ if and only if $f_D(n) \in \mathrm{NAF}(D)$. Further, if $n \in \mathrm{NAF}(D)$, then $R_D(n) = R_D(f_D(n)) \| g_D(n)$.*

Now, suppose $n \in \mathrm{NAF}(D)$. Then the finite string $R_D(n)$ can be computed with a finite number of recursive steps. This implies that there is some positive integer i such that ${f_D}^i(n) = 0$, which in turn implies that the procedure terminates. Conversely, suppose the procedure terminates. Then ${f_D}^i(n) = 0$ for some i, and clearly $0 \in \mathrm{NAF}(D)$. Thus, ${f_D}^i(n) \in \mathrm{NAF}(D)$, and by the lemma $n \in \mathrm{NAF}(D)$.

Procedure 4.5 is named EVAL$_\alpha$-$R_D(n)$. We justify this name by noting that if the procedure terminates, it returns a string with no leading zeros (i.e., $\widehat{\alpha}$) equal to $R_D(n)$. We are not able to evaluate $R_D(n)$ for all values of n using this procedure because we have not yet described a way to recognize when $R_D(n) = \perp$. We proceed to do this now.

To decide if $D = \{0, 1, x\}$ is an NADS, it suffices to determine if there are any $n \in \mathbb{Z}^+$ for which Procedure 4.5 fails to terminate. We can determine if the procedure will terminate by examining the iterates of f_D.

Let n be a positive integer. Observe that, for $n \not\equiv 3 \pmod 4$, we have that

$$n > f_D(n) \geq 0, \tag{4.3}$$

and, for $n \equiv 3 \pmod 4$, that

$$n > f_D(n) \iff n > -x/3, \tag{4.4}$$

$$f_D(n) \geq 0 \iff n \geq x. \tag{4.5}$$

Since x is negative, we see that any iterate of the function f_D, on input n, always results in a nonnegative integer. Consider the graph G_n having directed edges

$$n \to f_D(n) \to {f_D}^2(n) \to {f_D}^3(n) \to \cdots .$$

The vertices of G_n are nonnegative integers. Inequalities (4.3) and (4.4) tell us that there must be some vertex of G_n that is less than $\frac{-x}{3}$. Suppose ${f_D}^i(n) < \frac{-x}{3}$. We claim ${f_D}^{i+1}(n) < \frac{-x}{3}$ as well. This is clearly true if ${f_D}^i(n) \equiv 0, 1, 2 \pmod 4$. If ${f_D}^i(n) \equiv 3 \pmod 4$, then

$$\begin{aligned} {f_D}^i(n) < \frac{-x}{3} &\implies \frac{{f_D}^i(n) - x}{4} < \frac{\frac{-x}{3} - x}{4} \\ &\implies {f_D}^{i+1}(n) < \frac{-x - 3x}{12} = \frac{-x}{3}, \end{aligned}$$

and so the claim is true. The claim also tells us that if ${f_D}^i(n) < \frac{-x}{3}$, then any subsequent iterate of f_D must be less than $\frac{-x}{3}$.

From the preceding discussion it is clear that for a positive integer n, either

1. G_n is a path terminating at 0, or
2. G_n contains a directed cycle of integers in the interval $\{1, 2, \ldots, \lfloor \frac{-x}{3} \rfloor\}$.

If we can detect a directed cycle in G_n, then we can determine whether or not Procedure 4.5 will terminate on input n. To do this we need to compute and store some of the vertices of G_n. However, as Procedure 4.5 executes, it computes all the vertices of G_n, so we might as well modify the procedure to detect a directed cycle in G_n on its own. This modification is described below as Algorithm 4.7.

ALGORITHM 4.7. EVAL-$R_D(n)$

```
$\alpha \leftarrow \epsilon$
while $n > -x/3$
  do { $\alpha \leftarrow g_D(n) \parallel \alpha$
     { $n \leftarrow f_D(n)$
$\mathcal{S} \leftarrow \varnothing$
while $n \neq 0$
  do { if $n \in \mathcal{S}$
     {   then return $\perp$
     { $\mathcal{S} \leftarrow \mathcal{S} \cup \{n\}$
     { $\alpha \leftarrow g_D(n) \parallel \alpha$
     { $n \leftarrow f_D(n)$
return $\widehat{\alpha}$
```

Now we can use the title "Algorithm" rather than "Procedure" because EVAL-$R_D(n)$ terminates for every $n \in \mathbb{Z}^+$. (For some positive integers, it was shown that EVAL$_\alpha$-$R_D(n)$ fails to terminate, which is why it cannot technically be called an algorithm.) As its name suggests, Algorithm 4.7 evaluates $R_D(n)$ for any $n \in \mathbb{Z}^+$. It is possible to show that the running time of EVAL-$R_D(n)$ is $O(\lg n + |x|)$.

Returning to our main task of recognizing when $\{0, 1, x\}$ is an NADS, Algorithm 4.7 and the preceding analysis are very helpful since they lead us to the following result.

THEOREM 4.8. *Suppose x is a negative integer and $x \equiv 3 \pmod 4$. If every element in the set $\{n \in \mathbb{Z}^+ : n \leq \lfloor -x/3 \rfloor\}$ has a $\{0, 1, x\}$-NAF, then $\{0, 1, x\}$ is an NADS.*

Proof. From inspection of Algorithm 4.7 this result is almost immediate; however, we can give a formal argument using the graph G_n.

Suppose the hypothesis is true. We must argue that $\{0, 1, x\}$ is an NADS. Take any $n \in \mathbb{Z}^+$ and consider the graph G_n. Suppose G_n contains a directed cycle. Let n_0 be a vertex in this cycle. Then $1 \leq n_0 \leq \lfloor -x/3 \rfloor$, and G_{n_0} must contain the same directed cycle. This implies that n_0 does not have a $\{0, 1, x\}$-NAF, contrary to our hypothesis. So, G_n is a path terminating at 0, and thus n has a $\{0, 1, x\}$-NAF. □

Theorem 4.8 suggests a computational method of determining if $\{0, 1, x\}$ is an NADS. For each $n \in \mathbb{Z}^+, n \leq \lfloor -x/3 \rfloor$, compute EVAL-$R_D(n)$. If all of these values have $\{0, 1, x\}$-NAFs, then $\{0, 1, x\}$ is an NADS; otherwise, we find a value that does not have a $\{0, 1, x\}$-NAF, which proves that $\{0, 1, x\}$ is not an NADS. To recognize an NADS, this method requires $\lfloor -x/3 \rfloor$ calls to EVAL-$R_D(n)$. However, we can decrease this number, as the next result shows.

COROLLARY 4.9. *Suppose x is a negative integer and $x \equiv 3 \pmod 4$. If every element in the set $\{n \in \mathbb{Z}^+ : n \leq \lfloor -x/3 \rfloor, n \equiv 3 \pmod 4\}$ has a $\{0, 1, x\}$-NAF, then $\{0, 1, x\}$ is an NADS.*

Proof. If $\{0, 1, x\}$ is not an NADS, then choose the smallest integer $n_0 \in \mathbb{Z}^+$ such that G_{n_0} contains a directed cycle. By Theorem 4.8 it must be that $n_0 \leq \lfloor -x/3 \rfloor$. Let $n_1 = f_D(n_0)$; then (n_0, n_1) is an arc of G_n. If $n_0 \not\equiv 3 \pmod 4$, then $n_1 < n_0$ and G_{n_1} contains the same directed cycle, contrary to the choice of n_0. Thus, it must be that $n_0 \equiv 3 \pmod 4$. So, if the hypothesis is true, there can be no smallest positive integer which does not have a $\{0, 1, x\}$-NAF. Hence $\{0, 1, x\}$ is an NADS. □

Now we can detect an NADS of the form $\{0, 1, x\}$ with about $\lfloor -x/12 \rfloor$ calls to EVAL-$R_D(n)$. An optimized version of an algorithm utilizing this method is described in Algorithm 4.10. We have used this algorithm to find all the values of x greater than -10^6 such that $\{0, 1, x\}$ is an NADS; some of these values are listed in the appendix.

ALGORITHM 4.10. IS-NADS(x)

```
$[h]N \leftarrow 3$
$\mathcal{T} \leftarrow \varnothing$
while $N \leq -x/3$
  do $n \leftarrow N$
     $\mathcal{S} \leftarrow \varnothing$
     while $n \neq 0$ and $n \notin \mathcal{T}$
       do if $n \in \mathcal{S}$
            then return "no"
          $\mathcal{S} \leftarrow \mathcal{S} \cup \{n\}$
          $n \leftarrow f_D(n)$
     $N \leftarrow N + 4$
     $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{S}$
return "yes"
```

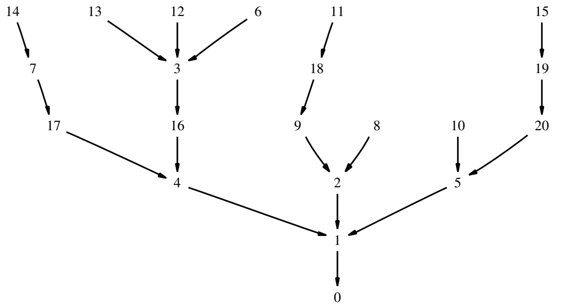


FIG. 5.1. $G(-61)$.

5. Directed graphs and NADS. For small values of x, a convenient way to demonstrate that $\{0, 1, x\}$ is an NADS is to draw a number of directed graphs. From the previous section, we know that $\{0, 1, x\}$ is an NADS if and only if each directed graph, G_n, $n \in \{1, 2, \dots, \lfloor \frac{-x}{3} \rfloor\}$, is a path terminating at zero. If we define

$$G(x) := \bigcup_{n=1}^{\lfloor \frac{-x}{3} \rfloor} G_n,$$

then we have that $\{0, 1, x\}$ is an NADS if and only if $G(x)$ is a directed tree rooted at zero. If $\{0, 1, x\}$ is not an NADS, then $G(x)$ must contain a directed cycle. In this section we discuss some of the properties of $G(x)$; in particular, we give a correspondence between strings in $\{00, 01, 0, 0x\}^*$, which represent nonzero multiples of Mersenne numbers and directed cycles of $G(x)$.

We start with an example. Let $x = -61$. Since $\lfloor \frac{-x}{3} \rfloor = 20$, $G(x)$ is the union of $G_1, G_2, \dots G_{20}$. A drawing of $G(x)$ is given in Figure 5.1. In the appendix, it is noted that $\{0, 1, -61\}$ is an NADS and, from Figure 5.1, we see that this is indeed the case since $G(x)$ contains no directed cycle.

The function g_D, which was defined in (4.2), can be used to label the arcs of each $G_1, G_2, \dots G_{20}$ as follows:

$$n \xrightarrow{g_D(n)} f_D(n) \xrightarrow{g_D(f_D(n))} {f_D}^2(n) \xrightarrow{g_D({f_D}^2(n))} {f_D}^3(n) \xrightarrow{g_D({f_D}^3(n))} \cdots .$$

Recall that g_D returns a string from the set $\{00, 01, 0, 0x\}$. These arc labels can be applied to $G(x)$, as shown in Figure 5.2.

The arc labels on this drawing of $G(x)$ allow us to easily determine the D-NAF of any node of $G(x)$. If n is a node then, since $G(x)$ is a tree, there is a unique directed path from n to the root node (i.e., G_n). The sequence of arc labels on the reverse of this path identifies the $\{0, 1, x\}$-NAF for n. For example, if we let $n = 14$, then from Figure 5.2 the directed path from 14 to 0 is

$$14 \xrightarrow{0} 7 \xrightarrow{0x} 17 \xrightarrow{01} 4 \xrightarrow{00} 1 \xrightarrow{01} 0.$$

If we read the sequence of arc labels above from right to left and concatenate them, we get the string $01\|00\|01\|0x\|0$. It is easily verified that $14 = (0100010x0)_2$.

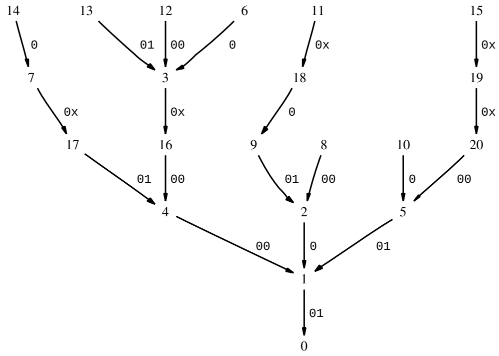


FIG. 5.2. $G(-61)$ *with arc labels.*

To see why this is true in general, suppose the path from n to 0 has length t and consider the label $g_D(n)$ on the arc $(n, f_D(n))$. From the definition of f_D and g_D, we have

$$\begin{aligned} f_D(n) &= \frac{n-(g_D(n))_2}{2^{|g_D(n)|}} \\ \Longrightarrow\ n &= 2^{|g_D(n)|} f_D(n) + (g_D(n))_2, \end{aligned} \tag{5.1}$$

where $|g_D(n)|$ denotes the length of the string $g_D(n)$. Replacing n with $f_D(n)$ in (5.1) we have

$$f_D(n) = 2^{|g_D(f_D(n))|} {f_D}^2(n) + (g_D(f_D(n)))_2. \tag{5.2}$$

Substituting (5.2) into (5.1) we find

$$\begin{aligned} n &= 2^{|g_D(f_D(n))|+|g_D(n)|} {f_D}^2(n) + 2^{|g_D(n)|}(g_D(f_D(n)))_2 + (g_D(n))_2 \\ \Longrightarrow\ n &= 2^{|g_D(f_D(n)) \| g_D(n)|} {f_D}^2(n) + (g_D(f_D(n)) \| g_D(n))_2. \end{aligned}$$

This method of substitution can be applied again. In (5.1), n can be replaced with ${f_D}^2(n)$, and then we can use this new equation to substitute for ${f_D}^2(n)$ above, and so on.

Let α be the string formed by concatenating the arc labels along the reverse of the path from n to 0. Then we have

$$\alpha = g_D({f_D}^{t-1}(n)) \| \cdots \| g_D({f_D}^2(n)) \| g_D(f_D(n)) \| g_D(n).$$

From (5.1), it follows that

$$n = 2^{|\alpha|} {f_D}^t(n) + (\alpha)_2. \tag{5.3}$$

Since the length of the path from n to 0 is t, ${f_D}^t(n) = 0$, and thus

$$n = (\alpha)_2,$$

that is, α is a D-NAF for n.

The main result of this section concerns directed cycles in $G(x)$, so let us consider an example that contains a directed cycle. Let $x = -41$. This value of x is not listed

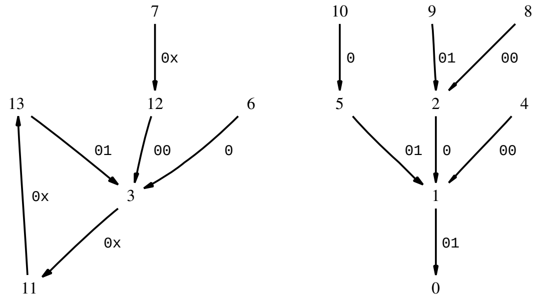


FIG. 5.3. *$G(-41)$ with arc labels.*

in the appendix, so we expect that $\{0, 1, -41\}$ is not an NADS, and the drawing of $G(x)$ in Figure 5.3 establishes this. Note that $G(x)$ consists of two components. Any node in the component of $G(x)$ that does not contain 0 does not have a D-NAF since there is no directed path from that node to 0.

Consider the directed cycle of $G(x)$. This cycle can be considered as a directed path from 3 to itself:

$$3 \xrightarrow{0x} 11 \xrightarrow{0x} 13 \xrightarrow{01} 3.$$

Reading the sequence of arc labels above from right to left and concatenating them, we get the string $01\|0x\|0x$. This string has length 6 and, because of this, we claim that $2^6 - 1$ must divide $(010x0x)_2$. Since $x = -41$, $(010x0x)_2 = -189$ and it is easy to check that this claim is valid. The following result provides an explanation.

THEOREM 5.1. *Suppose x is a negative integer and $x \equiv 3 \pmod 4$. Then, $G(x)$ has a directed cycle if and only if $\exists\, \alpha \in \{00, 01, 0, 0x\}^*$ such that $(\alpha)_2 \neq 0$ and $2^{|\alpha|} - 1 \mid (\alpha)_2$.*

Proof. Suppose $G(x)$ has a directed cycle. Choose a node n in some directed cycle of $G(x)$ and let t be the length of this cycle. Then we have

$$n \xrightarrow{g_D(n)} f_D(n) \xrightarrow{g_D(f_D(n))} {f_D}^2(n) \rightarrow \cdots \rightarrow {f_D}^{t-1}(n) \xrightarrow{g_D({f_D}^{t-1}(n))} n.$$

Some node in this cycle must be congruent to 3 modulo 4. If not, then the iterates of f_D are strictly decreasing on this cycle and we get

$$n > f_D(n) > {f_D}^2(n) > \cdots > {f_D}^{t-1}(n) > n,$$

which is a contradiction. A consequence of this fact is that one of the arcs in the cycle is labeled $0x$. As before, let

$$\alpha = g_D({f_D}^{t-1}(n)) \| \cdots \| g_D({f_D}^2(n)) \| g_D(f_D(n)) \| g_D(n).$$

Note that $(\alpha)_2 \neq 0$ because α contains the substring $0x$. Equation (5.3) gives us

$$n = 2^{|\alpha|} {f_D}^t(n) + (\alpha)_2.$$

Since ${f_D}^t(n) = n$, we have

$$\begin{aligned} n &= 2^{|\alpha|} n + (\alpha)_2 \\ \Longrightarrow -n(2^{|\alpha|} - 1) &= (\alpha)_2. \end{aligned}$$

Thus, $(\alpha)_2 \neq 0$ and $2^{|\alpha|} - 1 \mid (\alpha)_2$, as required.

Suppose $\alpha \in \{00, 01, 0, 0x\}^*$ has the property that $(\alpha)_2 \neq 0$ and $2^{|\alpha|} - 1 \mid (\alpha)_2$. The string $0x$ must be a substring of α; otherwise, $0 < (\alpha)_2 < 2^{|\alpha|} - 1$, and this contradicts our hypothesis that $2^{|\alpha|} - 1 \mid (\alpha)_2$. We claim that we can assume $(\alpha)_2$ is odd. To see why, let α' be any left cyclic shift of α. For some $u \in \mathbb{Z}^+$, we have

$$\begin{aligned} &(\alpha')_2 \equiv 2^u (\alpha)_2 \pmod{2^{|\alpha|} - 1} \\ \Longrightarrow\ &(\alpha')_2 \equiv 0 \pmod{2^{|\alpha|} - 1}, \end{aligned}$$

and since $|\alpha| = |\alpha'|$, this gives us that $2^{|\alpha'|} - 1 \mid (\alpha')_2$. Also, $(\alpha')_2 \neq 0$ because $(\alpha)_2 \neq 0$. Now, α contains the substring $0x$, so it must have some left cyclic shift that ends in 1 or x; that is, for some α', $(\alpha')_2$ is odd. Thus, if $(\alpha)_2$ is not odd, we can replace α with α', where $(\alpha')_2$ is odd.

Let $n = -\frac{(\alpha)_2}{2^{|\alpha|}-1}$. We will show that n is in a directed cycle of $G(x)$. Since α contains the substring $0x$, $|\alpha| \geq 2$, and so we have the following:

$$\begin{aligned} &-n(2^{|\alpha|} - 1) = (\alpha)_2 \\ \Longrightarrow\ & n = 2^{|\alpha|} n + (\alpha)_2 \\ \Longrightarrow\ & n \equiv (\alpha)_2 \pmod 4 \\ \Longrightarrow\ & \alpha = \alpha_1 \| g_D(n), \quad \text{where } \alpha_1 \in \{00, 01, 0, 0x\}^*. \end{aligned} \tag{5.4}$$

Using these implications, we can compute $f_D(n)$ as follows:

$$\begin{aligned} f_D(n) &= \frac{n - (g_D(n))_2}{2^{|g_D(n)|}} \\ &= \frac{2^{|\alpha|} n + (\alpha)_2 - (g_D(n))_2}{2^{|g_D(n)|}} \\ &= \frac{2^{|\alpha|} n + (\alpha_1 \| g_D(n))_2 - (g_D(n))_2}{2^{|g_D(n)|}} \\ &= 2^{|\alpha| - |g_D(n)|} n + (\alpha_1)_2 \\ &= 2^{|\alpha_1|} n + (\alpha_1)_2. \end{aligned} \tag{5.5}$$

Equation (5.5) is similar to (5.4). If $|\alpha_1| \geq 2$, the preceding arguments can be reapplied to compute ${f_D}^2(n)$. In doing so, we find

$${f_D}^2(n) = 2^{|\alpha_2|} n + (\alpha_2)_2,$$

where $\alpha_1 = \alpha_2 \| g_D(f_D(n))$ and $\alpha_2 \in \{00, 01, 0, 0x\}^*$. We can continue computing iterates of f_D in this manner until, for some $t \geq 1$, we obtain

$${f_D}^t(n) = 2^{|\alpha_t|} n + (\alpha_t)_2,$$

where $\alpha_{t-1} = \alpha_t \| g_D({f_D}^{t-1}(n))$, $\alpha_t \in \{00, 01, 0, 0x\}^*$, and $|\alpha_t| < 2$.

There are two cases to consider. If $|\alpha_t| = 0$, then it must be that $\alpha_t = \epsilon$, and thus

$${f_D}^t(n) = 2^0 n + (\epsilon)_2 = n.$$

Thus, n is in a directed cycle (of length t) in $G(x)$. If $|\alpha_t| = 1$, then it must be that $\alpha_t = 0$, and thus

$${f_D}^t(n) = 2^1 n + (0)_2 = 2n.$$

Recall that $(\alpha)_2$ is odd. Since $n = 2^{|\alpha|}n + (\alpha)_2$ and $|\alpha| \geq 2$, n is also odd. Thus, $2n \equiv 2 \pmod 4$ and so

$$f_D{}^{t+1}(n) = \frac{2n}{2} = n.$$

Thus, n is in a directed cycle (of length $t+1$) in $G(x)$. □

Theorem 5.1 gives a complete characterization of NADS; however, it is unclear if this characterization is helpful in finding values of x, which make $\{0, 1, x\}$ an NADS. On the other hand, Theorem 5.1 is very useful for finding values of x for which $\{0, 1, x\}$ is not an NADS. We give some examples of this in the next section.

The remainder of this paper reads as follows. In section 6, we give some infinite families of values for x for which D is *not* an NADS. In section 7, we give some infinite families of values for x for which D is an NADS. We conclude by mentioning some additional problems related to NADS in section 8.

6. Infinite families of non-NADS. Consider the list of x values that appears in the appendix. If we examine the first few entries of this list, we find no multiples of 3. In fact, this is true of the whole list, and the same can be said of multiples of 7 and 31. These observations are a consequence of the following result.

COROLLARY 6.1. *Let x be a negative integer with $x \equiv 3 \pmod 4$. If $(2^s - 1) \mid x$ for any $s \geq 2$, then $\{0, 1, x\}$ is not an NADS.*

Proof. This result follows from Theorem 5.1; however, it is just as easy to give a direct proof. Let $n = -x/(2^s - 1)$. We show that G_n contains a directed cycle. We have

$$\begin{aligned} & n(2^s - 1) \equiv -x \pmod 4 \\ \Longrightarrow\ & n(0 - 1) \equiv -3 \pmod 4 \\ \Longrightarrow\ & n \equiv 3 \pmod 4. \end{aligned}$$

Note that

$$n - x = \frac{-x}{2^s - 1} - x = \frac{-x - x(2^s - 1)}{2^s - 1} = 2^s \frac{-x}{2^s - 1} = 2^s n.$$

Now,

$$f_D(n) = \frac{n - x}{4} = 2^{s-2}n.$$

Subsequent iterates of f_D will cancel out the factor 2^{s-2}. Thus, for some i, $f_D{}^i(n) = n$ and so G_n contains a directed cycle. □

Corollary 6.1 says that many sets $\{0, 1, x\}$ are not NADS. In particular, it rules out sets in which x is divisible by 3, 7, 31, etc. Besides numbers of the form $2^s - 1$, $s \geq 2$, there are many other *nonallowable factors* of x. For example, if any of the integers

$$73, 85, 89, 337, 451, 1103, 1205, 1285, 2089$$

divides x, then it is possible to show that, for a carefully chosen value of n, G_n contains a directed cycle. This proof technique is not fully satisfying since it does little to elucidate why one integer is a nonallowable factor and another is not. A better approach is presented in the following corollary to Theorem 5.1.

COROLLARY 6.2. *Suppose x_0 is an integer. If $\exists\,\beta \in \{00, 0, 0x_0\}^*$ such that $(\beta)_2 \neq 0$ and $2^{|\beta|} - 1 \mid (\beta)_2$, then x_0 is a nonallowable factor.*

Proof. Notice there are no restrictions put on the integer x_0. Let x be a negative integer with $x \equiv 3 \pmod 4$ and $x_0 \mid x$. We must show that $\{0, 1, x\}$ is not an NADS. Let α be the string formed by changing every occurrence of x_0 in β to x. It is easy to see that $(\alpha)_2 = \frac{x}{x_0}(\beta)_2$, $\alpha \in \{00, 0, 0x\}^*$, and $|\alpha| = |\beta|$. Now,

$$\begin{aligned} & 2^{|\beta|} - 1 \mid (\beta)_2 \\ \Longrightarrow\ & 2^{|\beta|} - 1 \mid \frac{x}{x_0}(\beta)_2 \\ \Longrightarrow\ & 2^{|\beta|} - 1 \mid (\alpha)_2 \\ \Longrightarrow\ & 2^{|\alpha|} - 1 \mid (\alpha)_2. \end{aligned}$$

Since $\alpha \in \{00, 01, 0, 0x\}^*$ and $(\alpha)_2 \neq 0$, by Theorem 5.1 we have that $\{0, 1, x\}$ is not an NADS. □

We can use this result to generate nonallowable factors. All we need to do is find an integer x_0 and a string $\beta \in \{00, 0, 0x_0\}^*$, where β is not an all-zero string, such that $2^{|\beta|} - 1 \mid (\beta)_2$. To do this we first choose a string $\beta' \in \{00, 0, 01\}^*$ that is not an all-zero string. Now, we find an integer x_0 such that $2^{|\beta'|} - 1 \mid x_0(\beta')_2$. The smallest positive value of x_0 that satisfies this relation is

$$\frac{2^{|\beta'|} - 1}{\gcd(2^{|\beta'|} - 1, (\beta')_2)}.$$

We assign x_0 this value. If we change each occurrence of 1 in the string β' to x_0, we get a string $\beta \in \{00, 0, 0x_0\}^*$ such that $(\beta)_2 \neq 0$ and $2^{|\beta|} - 1 \mid (\beta)_2$. So, by the corollary, x_0 is a nonallowable factor. Here is a short example. Let $\beta' = 000010101$. Then $|\beta'| = 9$, $(\beta')_2 = 21$, and so

$$x_0 = \frac{2^9 - 1}{\gcd(2^9 - 1, 21)} = 73.$$

Thus, 73 is a nonallowable factor.

More generally, Theorem 5.1 can be used to generate infinite families of non-NADS, which do not necessarily involve nonallowable factors. We know $\{0, 1, x\}$ is not an NADS if we can find a string $\alpha \in \{00, 01, 0, 0x\}^*$ such that $-n(2^{|\alpha|} - 1) = (\alpha)_2$. If we fix α and solve the resulting integer equation for x, this will give us an infinite family of non-NADS. For example, suppose we fix $\alpha = 01010x0x$; then

$$\begin{aligned} & -n(2^{|\alpha|} - 1) = (01010x0x)_2 \\ \Longleftrightarrow\ & -n(2^8 - 1) = (01010000)_2 + x(00000101)_2 \\ \Longleftrightarrow\ & -255n = 80 + 5x \\ \Longleftrightarrow\ & -51n = 16 + x. \end{aligned}$$

Thus, if $x \equiv -16 \pmod{51}$, then $\{0, 1, x\}$ cannot be an NADS.

Some of our first results on infinite families of non-NADS, which were discovered empirically, are unified as corollaries of Theorem 5.1. The following two results demonstrate this.

COROLLARY 6.3. *If $\frac{3-x}{4} = 11 \cdot 2^i$, where $i \geq 0$, then $\{0, 1, x\}$ is not an NADS.*

Proof. We have,

$$\begin{aligned} &\frac{3-x}{4} = 11 \cdot 2^i \\ \Longrightarrow\ &3 - x = 11 \cdot 2^{i+2} \\ \Longrightarrow\ &11 - x = 11 \cdot 2^{i+2} + 8 \\ \Longrightarrow\ &-11(2^{i+2} - 1) = 8 + x \\ \Longrightarrow\ &-11(2^{i+2} - 1) = (0100x)_2. \end{aligned}$$

The length of the string $0100x$ is 5. If $i + 2 \geq 5$, we can prepend zeros to $0100x$ and build a string α such that $|\alpha| = i + 2$; thus, by Theorem 5.1 we are done. If $i + 2 < 5$, it must be that $i = 0, 1, 2$.

When $i = 0$, $x = -41$ and from the drawing in Figure 5.3 we see $G(-41)$ has a directed cycle. When $i = 1$, $x = -85$ and then G_3 is a directed cycle:

$$3 \to 22 \to 11 \to 24 \to 6 \to 3.$$

When $i = 2$, $x = -173$ and G_3 is also a directed cycle:

$$3 \to 44 \to 11 \to 46 \to 23 \to 49 \to 12 \to 3.$$

In any case, $\{0, 1, x\}$ is not an NADS, as required. □

Corollary 6.4. *Let $\frac{3-x}{4} = 7 \cdot 2^i$, where $i \geq 0$. Then $\{0, 1, x\}$ is an NADS if and only if $i \in \{0, 1\}$.*

Proof. We have

$$\begin{aligned} &\frac{3-x}{4} = 7 \cdot 2^i \\ \Longrightarrow\ &3 - x = 7 \cdot 2^{i+2} \\ \Longrightarrow\ &7 - x = 7 \cdot 2^{i+2} + 4 \\ \Longrightarrow\ &-7(2^{i+2} - 1) = 4 + x \\ \Longrightarrow\ &-7(2^{i+2} - 1) = (010x)_2. \end{aligned}$$

Arguing as in the previous corollary, if $i + 2 \geq 4$, then by Theorem 5.1, $\{0, 1, x\}$ is not an NADS. If $i + 2 < 4$, it must be that $i = 0, 1$.

When $i = 0$, $x = -25$ and when $i = 1$, $x = -53$. By drawing the graphs $G(-25)$ and $G(-53)$, it is easy to verify that both of these values give NADSs (this is confirmed in the appendix). □

Not all infinite families of non-NADS are derived from Theorem 5.1. Consider the set of integers $\mathrm{NAF}(\{0, 1\})$. If this set is ordered from smallest to largest, we sometimes notice large gaps between consecutive elements. One type of gap is described as follows. For $i \geq 0$, let

$$m_i := \left\lfloor \frac{2^{i+1} - 1}{3} \right\rfloor.$$

Computing the first few values of m_i, we have

i	m_i	
0	0	
1	1	$= (1)_2$
2	2	$= (10)_2$
3	5	$= (101)_2$
4	10	$= (1010)_2$
5	21	$= (10101)_2$
6	42	$= (101010)_2$
7	85	$= (1010101)_2$
$\vdots$	$\vdots$	

It is easy to see that if $a \in \mathrm{NAF}(\{0,1\})$, then it is never true that $m_i < a < 2^i$. This observation gives us another infinite family.

THEOREM 6.5. *Let x be an integer such that $4m_i - 1 < -x < 3 \cdot 2^i$ for some $i \geq 0$. If there exists $n \in \{1, 2, \ldots, \lfloor -x/3 \rfloor\}$ with $n \equiv 3 \pmod 4$, then $\{0,1,x\}$ is not an NADS.*

Proof. We can assume $x \equiv 3 \pmod 4$ since, otherwise, $\{0,1,x\}$ cannot be an NADS. Suppose to the contrary that $\{0,1,x\}$ is an NADS. Then, in the graph $G(x)$, there must be a directed path from n to 0. Let n_0 be the integer on this path that is *closest* to 0 and is congruent to 3 modulo 4. The arc labels on the path from n_0 to 0 give the $\{0,1,x\}$-NAF for n_0. It must be that $n_0 = (\alpha \| 0x)_2$ with $\alpha \in \{00, 01, 0\}^*$ (if α contained the substring $0x$, this would contradict our choice of n_0).

Now,

$$\begin{aligned}
& 1 \leq n_0 \leq -x/3 \\
\Longrightarrow\ & 1 \leq (\alpha \| 0x)_2 \leq -x/3 \\
\Longrightarrow\ & 1 \leq 4(\alpha)_2 + x \leq -x/3 \\
\Longrightarrow\ & \frac{1-x}{4} \leq (\alpha)_2 \leq \frac{-x/3 - x}{4} \\
\Longrightarrow\ & \frac{1-x}{4} \leq (\alpha)_2 \leq -x/3.
\end{aligned}$$

By hypothesis, we have

$$\begin{aligned}
& 4m_i - 1 < -x \quad \text{and} \quad -x < 3 \cdot 2^i \\
\Longrightarrow\ & m_i < \frac{1-x}{4} \quad \text{and} \quad \frac{-x}{3} < 2^i.
\end{aligned}$$

Thus, for some $i \geq 0$, we have

$$m_i < (\alpha)_2 < 2^i,$$

which is a contradiction. Thus, $\{0,1,x\}$ is not an NADS. □

For example, if $i = 5$, then $-(4m_5 - 1) = -83$ and $-3 \cdot 2^5 = -96$. Theorem 6.5 tells us that no value of x with $-83 < x < -96$ can give an NADS. In addition, the proof of Theorem 6.5 also gives us some information about the graphs $G(x)$ for such values of x. For each of these graphs, in the component that contains 0 there can be no integer congruent to 3 modulo 4 (or, equivalently, no arc label in this component can be $0x$). This property can be observed in $G(-85)$, which is drawn in Figure 6.1.

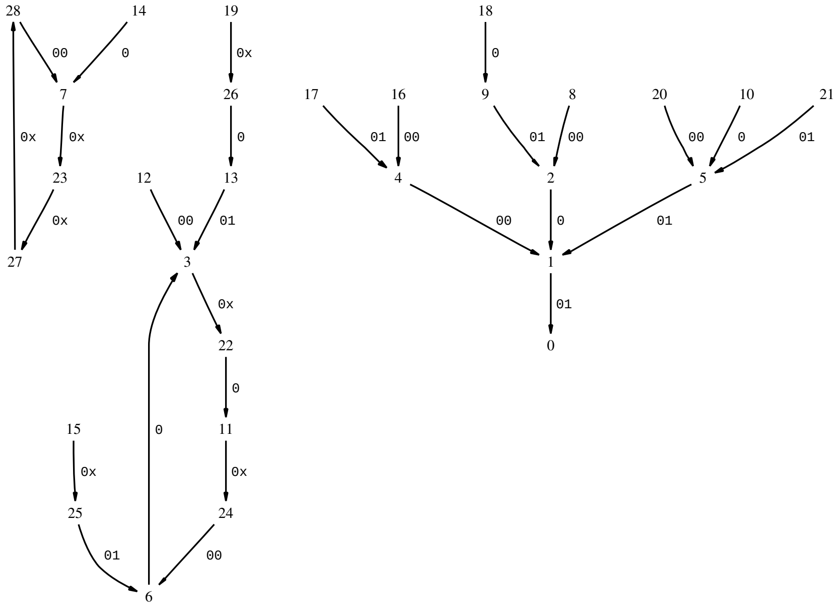


FIG. 6.1. *$G(-85)$ with arc labels.*

7. Infinite families of NADS. If n is a nonnegative integer, $w(n)$ denotes the number of ones in the usual $\{0, 1\}$-radix 2 representation of n (i.e., the Hamming weight of n). We use the function $w(n)$ to describe two infinite families.

THEOREM 7.1. *Let x be a negative integer with $x \equiv 3 \pmod 4$. If $w(\frac{3-x}{4}) = 1$, then $\{0, 1, x\}$ is an NADS.*

Proof. Suppose $\{0, 1, x\}$ is not an NADS. Then there is some $n \in \mathbb{Z}^+$ for which the graph G_n contains a directed cycle. We can assume n is a vertex of this cycle. Let t be the number of vertices in the cycle; then

$$n \to f_D(n) \to {f_D}^2(n) \to \cdots \to {f_D}^{t-1}(n) \to n.$$

Let $n' = f_D(n)$. We want to relate $w(n')$ to $w(n)$. There are four possible residues of n modulo 4 and, for the residues $0, 1, 2$, we can determine $w(n')$ exactly as follows:

$n \bmod 4$	n'	$w(n')$
0	$\frac{n}{4}$	$w(n)$
1	$\frac{n-1}{4}$	$w(n) - 1$
2	$\frac{n}{2}$	$w(n)$

If $n \equiv 3 \pmod 4$, we have

$$n' = \frac{n-x}{4} = \frac{n-3}{4} + \frac{3-x}{4}.$$

By hypothesis, $w(\frac{3-x}{4}) = 1$, and so

$$\begin{aligned} w(n') &= w\left(\frac{n-3}{4} + \frac{3-x}{4}\right) \\ &\le w\left(\frac{n-3}{4}\right) + w\left(\frac{3-x}{4}\right) \\ &= w(n) - 2 + 1 \\ &= w(n) - 1. \end{aligned}$$

So, in any case, $w(n') \le w(n)$, but if n is odd, then we have the strict inequality $w(n') < w(n)$. Applying this inequality to the integers in the cycle of G_n, we see that

$$w(n) \ge w(f_D(n)) \ge w({f_D}^2(n)) \ge \cdots \ge w({f_D}^{t-1}(n)) \ge w(n).$$

However, some vertex in this cycle must be congruent to 3 modulo 4. If not, then the iterates of f_D are strictly decreasing on this cycle and we get

$$n > f_D(n) > {f_D}^2(n) > \cdots > {f_D}^{t-1}(n) > n,$$

which is a contradiction. So, there is some odd vertex in the cycle, which means one of the inequalities relating the Hamming weights of adjacent vertices is strict. This implies that $w(n) > w(n)$, which is a contradiction.

So, G_n cannot contain a directed cycle, and hence $\{0, 1, x\}$ is an NADS. □

When x is negative, $w(\frac{3-x}{4}) = 1$ if and only if $\frac{3-x}{4} = 2^t$, $t \ge 0$. Letting $t = 0, 1, 2, 3, 4, \ldots$ we see that Theorem 7.1 asserts that $x = -1, -5, -13, -29, -61, \ldots$ all yield NADS. Our next result also describes an infinite family using the function $w(n)$. However, when compared to the previous result, proving that $\{0, 1, x\}$ is an NADS for each x in this second infinite family seems to be more difficult.

Theorem 7.2. *Let x be a negative integer with $x \equiv 3 \pmod 4$. If $w(\frac{3-x}{4}) = 2$ and $2^s - 1$ does not divide x for any $s \in \mathbb{Z}^+$, $s \ge 2$, then $\{0, 1, x\}$ is an NADS.*

To prove this result we suppose x is a negative integer with $x \equiv 3 \pmod 4$ and $w(\frac{3-x}{4}) = 2$. We will argue that if $\{0, 1, x\}$ is not an NADS, then it must be that $2^s - 1$ divides x for some $s \in \mathbb{Z}^+$, $s \ge 2$.

We begin our argument following the proof of Theorem 7.1. Suppose $\{0, 1, x\}$ is not an NADS. Then there is some $n \in \mathbb{Z}^+$ for which the graph G_n contains a directed cycle. We can assume n is a vertex of this cycle and, as described in section 5, we can label the arcs of this cycle using the function g_D. Let t be the number of vertices in the cycle; then

$$n \xrightarrow{g_D(n)} f_D(n) \xrightarrow{g_D(f_D(n))} {f_D}^2(n) \to \cdots \to {f_D}^{t-1}(n) \xrightarrow{g_D({f_D}^{t-1}(n))} n.$$

Let $n' = f_D(n)$. We want to relate $w(n')$ to $w(n)$. There are four possible residues of n modulo 4, and for the residues $0, 1, 2$ we can determine $w(n')$ exactly as follows:

$n \bmod 4$	n'	$w(n')$
0	$\frac{n}{4}$	$w(n)$
1	$\frac{n-1}{4}$	$w(n) - 1$
2	$\frac{n}{2}$	$w(n)$

If $n \equiv 3 \pmod 4$, we have

$$n' = \frac{n-x}{4} = \frac{n-3}{4} + \frac{3-x}{4}.$$

By hypothesis, $w(\frac{3-x}{4}) = 2$, and so

$$\begin{aligned} w(n') &= w\left(\frac{n-3}{4} + \frac{3-x}{4}\right) \\ &\leq w\left(\frac{n-3}{4}\right) + w\left(\frac{3-x}{4}\right) \\ &= w(n) - 2 + 2 \\ &= w(n). \end{aligned}$$

So, in any case, $w(n') \leq w(n)$, but if $n \equiv 1 \pmod 4$, then we have the strict inequality $w(n') < w(n)$. Applying this inequality to the integers in the cycle of G_n, we see that

$$w(n) \geq w(f_D(n)) \geq w({f_D}^2(n)) \geq \cdots \geq w({f_D}^{t-1}(n)) \geq w(n).$$

No vertex in this cycle can be congruent to 1 modulo 4; otherwise, one of the inequalities above would be strict and this would imply $w(n) > w(n)$, which is a contradiction. Also, at least one vertex in this cycle is congruent to 3 modulo 4; otherwise, by definition of f_D, the vertices would form a strictly decreasing integer sequence, which would imply $n > n$, which is a contradiction.

Let α be the string formed by concatenating all of the arc labels from the cycle:

$$\alpha = g_D({f_D}^{t-1}(n)) \| \cdots \| g_D({f_D}^2(n)) \| g_D(f_D(n)) \| g_D(n).$$

Since α is a concatenation of strings from the set $\{00, 0, 0x\}$, it is nonadjacent and, further, for the same reason, every cyclic shift of α is also nonadjacent (i.e., α is *cyclically nonadjacent*). Equation (5.3) from section 5 tells us

$$n = 2^{|\alpha|} {f_D}^t(n) + (\alpha)_2.$$

Since ${f_D}^t(n) = n$, we have

$$n = 2^{|\alpha|} n + (\alpha)_2. \tag{7.1}$$

The integer $(\alpha)_2$ is divisible by x. Let

$$A = \frac{(\alpha)_2}{x} \quad \text{and} \quad a = |\alpha|.$$

From (7.1) we have

$$-xA \equiv 0 \pmod{2^a - 1}. \tag{7.2}$$

Since $w(\frac{3-x}{4}) = 2$, for some $u, v \in \mathbb{Z}$ we have

$$-x = 2^u + 2^v - 3, \quad u > v \geq 2,$$

and now (7.2) implies

$$(2^u + 2^v - 3)A \equiv 0 \pmod{2^a - 1}, \quad \text{where } u > v \geq 2. \tag{7.3}$$

To finish the proof we need a lemma. Before we can introduce the lemma, we need a definition.

DEFINITION 7.3. *An integer $B \in \mathbb{Z}$ is* length-ℓ cyclically nonadjacent *(ℓ-CNA) if $B \neq 0$, and there is a cyclically nonadjacent string $\beta \in \{0,1\}^\ell$ such that $(\beta)_2 = B$.*

Note that, in this definition, the string β may have leading zeros. For example, 21 is length-6 cyclically nonadjacent (6-CNA) since the string $010101 \in \{0,1\}^6$ is cyclically nonadjacent and $(010101)_2 = 21$. However, 21 is not 5-CNA because the only string in $\{0,1\}^5$ giving a representation of 21 is 10101, but the cyclic shift 01011 of this string is not nonadjacent. Now we are ready for the lemma.

LEMMA 7.4. *If B is ℓ-CNA and the congruence*

$$(2^u + 2^v - 3)B \equiv 0 \pmod{2^\ell - 1}$$

holds for some $u, v \in \mathbb{Z}$, $u > v \geq 2$, then either

$$\gcd(u, v-1) > 1 \quad \textit{or} \quad \gcd(u-1, v) > 1.$$

Assuming, for the moment, that Lemma 7.4 is true, our proof of Theorem 7.2 continues as follows. The string α is CNA; therefore so is the string formed by changing each occurrence of x in α to 1. This establishes that A is a-CNA, because $A = \frac{(\alpha)_2}{x}$. Now we can apply Lemma 7.4 to (7.3) and deduce, without loss of generality, that $\gcd(u, v-1) > 1$. Let $s = \gcd(u, v-1)$. Note that

$$-x = 2^u + 2^v - 3 = (2^u - 1) + 2(2^{v-1} - 1).$$

Since $\gcd(2^u - 1, 2^{v-1} - 1) = 2^{\gcd(u,v-1)} - 1 = 2^s - 1$, we have that $2^s - 1 \mid x$, where $s \in \mathbb{Z}^+$ and $s \geq 2$, which is exactly what we wanted to show. (If x was chosen so as to satisfy all the conditions of Theorem 7.2, then $2^s - 1$ cannot divide x, and thus it must be that $\{0, 1, x\}$ is an NADS.) This concludes our proof of Theorem 7.2; however, we still have to deal with Lemma 7.4.

In proving Lemma 7.4, we will use the following easy result.

LEMMA 7.5. *For any two nonempty subsets $S, T \subseteq \{0, 1, \ldots, \ell - 1\}$,*

$$\sum_{s \in S} 2^s \equiv \sum_{t \in T} 2^t \pmod{2^\ell - 1}$$

if and only if $S = T$.

Proof. We have $0 < \sum_{s \in S} 2^s \leq 2^\ell - 1$, and $0 < \sum_{t \in T} 2^t \leq 2^\ell - 1$. Thus,

$$\begin{aligned} &\sum_{s \in S} 2^s \equiv \sum_{t \in T} 2^t \pmod{2^\ell - 1} \\ \iff &\sum_{s \in S} 2^s = \sum_{t \in T} 2^t \\ \iff &S = T. \qquad \square \end{aligned}$$

Proof of Lemma 7.4. We fix some notation that will help describe our proof of Lemma 7.4. From now on, we let B be an integer satisfying the hypothesis of Lemma 7.4. B is ℓ-CNA and we let $\beta = b_{\ell-1} \ldots b_1 b_0$ be the string in $\{0,1\}^\ell$, which establishes this. Further, let $S = \{i : b_i = 1\}$. For $k \in \mathbb{Z}$, define

$$S + k = \{(s + k) \bmod \ell : s \in S\}.$$

The set $S+k$ is called a *translate* of S modulo ℓ. Using this notation, we have

$$\begin{aligned} &(2^u+2^v-3)B \equiv 0 \pmod{2^\ell-1} \\ \iff &(2^u+2^v)B \equiv 3B \pmod{2^\ell-1} \\ \iff &(S+u)\cup(S+v) = (S+1)\cup S, \end{aligned} \tag{7.4}$$

where the last equivalence follows from the fact that B is ℓ-CNA and from Lemma 7.5. Because B is ℓ-CNA, the union on the right-hand side of (7.4), and hence also the left-hand side, is disjoint. We will establish Lemma 7.4 by analyzing this set equality.

We need one more concept. The *cyclic order* of B is the smallest positive integer k such that

$$2^kB \equiv B \pmod{2^\ell-1}.$$

We denote this integer by $\overset{\hookrightarrow}{\operatorname{ord}}(B)$. Such an integer always exists since

$$2^\ell B \equiv B \pmod{2^\ell-1}.$$

Using the quotient-remainder theorem, it is easy to show that for any $m\in\mathbb{Z}^+$,

$$2^mB \equiv B \pmod{2^\ell-1} \iff \overset{\hookrightarrow}{\operatorname{ord}}(B) \mid m.$$

Applying this result, we see that $\overset{\hookrightarrow}{\operatorname{ord}}(B)\mid\ell$. (An equivalent definition of $\overset{\hookrightarrow}{\operatorname{ord}}(B)$ can be made by considering the string β. The smallest number of left cyclic shifts that, when applied to β, results in the string β is exactly $\overset{\hookrightarrow}{\operatorname{ord}}(B)$.)

We claim that we can assume $\overset{\hookrightarrow}{\operatorname{ord}}(B)=\ell$ in the hypotheses of Lemma 7.4. We justify this claim as follows. Let $k=\overset{\hookrightarrow}{\operatorname{ord}}(B)$ and suppose $k<\ell$. Since $k\mid\ell$ we can write $km=\ell$ for some positive integer m. Since B is ℓ-CNA we have

$$B = (2^{(m-1)k}+\cdots+2^{2k}+2^k+1)B' = \frac{2^\ell-1}{2^k-1}B',$$

where $B'=(b_{k-1}\dots b_1b_0)_2$, and B' is k-CNA. Now, for any positive integer j, we have

$$\begin{aligned} &2^jB \equiv B \pmod{2^\ell-1} \\ \iff\ &2^j\frac{2^\ell-1}{2^k-1}B' \equiv \frac{2^\ell-1}{2^k-1}B' \left(\operatorname{mod}\ \frac{2^\ell-1}{2^k-1}2^k-1\right) \\ \iff\ &2^jB' \equiv B' \pmod{2^k-1}, \end{aligned}$$

and so it must be that $\overset{\hookrightarrow}{\operatorname{ord}}(B')=k$ (i.e., $\overset{\hookrightarrow}{\operatorname{ord}}(B')$ is as large as possible). Also, we have

$$\begin{aligned} &(2^u+2^v-3)B \equiv 0 \pmod{2^\ell-1} \\ \iff\ &(2^u+2^v-3)\frac{2^\ell-1}{2^k-1}B' \equiv 0 \left(\operatorname{mod}\ \frac{2^\ell-1}{2^k-1}2^k-1\right) \\ \iff\ &(2^u+2^v-3)B' \equiv 0 \pmod{2^k-1}. \end{aligned}$$

So if we can prove Lemma 7.4 for all B with $\overset{\hookrightarrow}{\operatorname{ord}}(B)$ as large as possible, then by the above arguments, it is true for all B.

Returning to the set equality described in (7.4), recall that $S \subseteq \{0, 1, \ldots, \ell - 1\}$. Since S is a subset of integers, its elements can be ordered from smallest to largest. From S we define a sequence, $d(S)$, of differences modulo ℓ,

$$d(S) := (s_1 - s_0, s_2 - s_1, \ldots, s_{p-1} - s_{p-2}, s_0 - s_{p-1}),$$

where

$$S = \{s_0, s_1, \ldots, s_{p-1}\} \quad \text{with } s_0 < s_1 < \cdots < s_{p-1}.$$

Because B is ℓ-CNA, each of the differences in the sequence $d(S)$ must be at least 2. The definition of $d(S)$ can be extended to the translates of S. For any $k \in \mathbb{Z}$, $S + k$ can be considered as a subset of $\{0, 1, \ldots, \ell - 1\}$, and hence it can also be ordered from smallest to largest. Thus, $d(S + k)$ can be defined in the same way as $d(S)$. It is easy to show that $d(S + k)$ is a cyclic shift of $d(S)$. Because of this property there are at most p different sequences of the form $d(S + k)$, where $p = |S|$. In fact, we can show there are exactly p such sequences.

Let

$$t_i := \ell - s_i \quad \text{for } 0 \leq i \leq p - 1.$$

The smallest element in each of the translates $S + t_0, S + t_1, \ldots, S + t_{p-1}$ is equal to 0. Thus, for $i, j \in \{0, 1, \ldots, p - 1\}$, we have

$$d(S + t_i) = d(S + t_j) \iff S + t_i = S + t_j.$$

Let $i \geq j$. Then we have

$$\begin{aligned}
&S + t_i = S + t_j \\
\iff &2^{t_i} B \equiv 2^{t_j} B \pmod{2^\ell - 1} \\
\iff &2^{t_i - t_j} B \equiv B \pmod{2^\ell - 1} \\
\iff &\overset{\hookrightarrow}{\mathrm{ord}}(B) \mid (t_i - t_j) \\
\iff &\ell \mid (t_i - t_j) \\
\iff &t_i = t_j.
\end{aligned}$$

So, each of the sequences $d(S + t_0), d(S + t_1), \ldots, d(S + t_{p-1})$ is distinct, and hence there are exactly p different sequences of the form $d(S + k)$, where $k \in \mathbb{Z}$.

By applying a lexicographical ordering to the sequences $d(S+t_0), d(S+t_1), \ldots, d(S+t_{p-1})$ we can identify a *unique* smallest sequence. Let t^* be the value of t_i which corresponds to this smallest sequence. Note that

$$\begin{aligned}
&(S + u) \cup (S + v) = (S + 1) \cup S \\
\iff &((S + u) \cup (S + v)) + t^* = ((S + 1) \cup S) + t^* \qquad (7.5)\\
\iff &(S + u + t^*) \cup (S + v + t^*) = (S + 1 + t^*) \cup (S + t^*).
\end{aligned}$$

We have $0 \in S + t^*$, so either $0 \in S + u + t^*$ or $0 \in S + v + t^*$. Without loss of generality we can assume $0 \in S + v + t^*$. We will show that $S + v + t^* = S + t^*$.

Let

$$d(S + t^*) = (d_0, d_1, d_2, \ldots, d_{p-1}),$$

and note that

$$S + t^* = \{0, d_0, d_1 + d_0, d_2 + d_1 + d_0, \dots\}.$$

Also, let

$$d(S + u + t^*) = (u_0, u_1, u_2, \dots, u_{p-1}),$$
$$d(S + v + t^*) = (v_0, v_1, v_2, \dots, v_{p-1}).$$

Since $d(S + t^*)$ is a lexicographically smallest sequence of the form $d(S + k)$, where $k \in \mathbb{Z}$, we have

$$d(S + t^*) \leq d(S + u + t^*) \quad \text{and} \quad d(S + t^*) \leq d(S + v + t^*).$$

Recall $0 \in S + t^*$ and $0 \in S + v + t^*$. Since $0 \in S + t^*$, we have $1 \in S + 1 + t^*$. By (7.5), either $1 \in S + u + t^*$ or $1 \in S + v + t^*$. Suppose $1 \in S + v + t^*$. Then both 0 and 1 are elements of $S + v + t^*$. No two elements in any translate of S can have a difference of 1; otherwise, this contradicts the fact that B is ℓ-CNA. So, it must be that $1 \in S + u + t^*$.

We now know the smallest elements in each of the sets $S + u + t^*$, $S + v + t^*$, $S + 1 + t^*$, $S + t^*$. The next smallest element of $S + t^*$ is d_0. Again, by (7.5), either $d_0 \in S + u + t^*$ or $d_0 \in S + v + t^*$. Suppose $d_0 \in S + u + t^*$. Then, since both 1 and d_0 are in $S + u + t^*$ and 1 is the smallest element of this set, we have

$$u_0 \leq d_0 - 1 < d_0.$$

However, $d(S + t^*) \leq d(S + u + t^*)$ implies that $d_0 \leq u_0$, which gives a contradiction. So, it must be that $d_0 \in S + v + t^*$, and hence, $d_0 + 1 \in S + u + t^*$.

From our lexicographical ordering we have $d_0 \leq v_0$. Since the smallest element of $S + v + t^*$ is 0 and d_0 is also in this set, we have

$$v_0 \leq d_0 - 0 = d_0.$$

Hence, $v_0 = d_0$. Similarly,

$$d_0 \leq u_0 \quad \text{and} \quad u_0 \leq (d_0 + 1) - 1 = d_0,$$

and so $u_0 = d_0$. From these two equalities, we have that d_0 and $d_0 + 1$ are the second smallest elements of the sets $S + v + t^*$ and $S + u + t^*$, respectively. Further, our lexicographical ordering now implies that $d_1 \leq v_1$ and $d_1 \leq u_1$.

The next smallest element of $S + t^*$ is $d_1 + d_0$. Either $d_1 + d_0 \in S + u + t^*$ or $d_1 + d_0 \in S + v + t^*$. Suppose $d_1 + d_0 \in S + u + t^*$. This implies that

$$u_1 \leq (d_1 + d_0) - (d_0 + 1) = d_1 - 1 < d_1,$$

which is a contradiction. So, $d_1 + d_0 \in S + v + t^*$, and hence, $d_1 + d_0 + 1 \in S + u + t^*$.

We now have

$$d_1 \leq v_1 \quad \text{and} \quad v_1 \leq (d_1 + d_0 + 1) - (d_0 + 1) = d_1,$$

so $v_1 = d_1$. Also

$$d_1 \leq u_1 \quad \text{and} \quad u_1 \leq (d_1 + d_0) - d_0 = d_1,$$

and so $u_1 = d_1$. Thus we can identify the third smallest elements of the sets $S+v+t^*$ and $S+u+t^*$. Further, we have that $d_2 \leq v_2$ and $d_2 \leq u_2$.

By repeating the previous arguments, we can show that each element of $S+t^*$, from smallest to largest, must also be an element of $S+v+t^*$. Thus, $S+v+t^* = S+t^*$ and so $S+v = S$. In (7.4), the union operations are both disjoint, and hence $S+v=S$ implies $S+u = S+1$. Now,

$$\begin{aligned} & S+v = S \\ \Longrightarrow\ & 2^v B \equiv B \pmod{2^\ell - 1} \\ \Longrightarrow\ & \overset{\hookrightarrow}{\mathrm{ord}}(B) \mid v \\ \Longrightarrow\ & \ell \mid v. \end{aligned}$$

Similarly, $\ell \mid (u-1)$. Thus $\gcd(u-1, v) \geq \ell > 1$. This proves the lemma. □

Looking at an example can help us connect the different steps in the proof of Theorem 7.2. Suppose $x = 3 - (2^u + 2^v)$ with $u > v \geq 2$. If $\{0,1,x\}$ is not an NADS, then $\exists\, \alpha \in \{00, 01, 0, 0x\}^*$ such that $(\alpha)_2 \equiv 0 \pmod{2^{|\alpha|} - 1}$. By the definition of x, it must be that $\alpha \in \{00, 0, 0x\}^*$. We will suppose $\alpha = 0x0x000x0x0x000x$, and so $|\alpha| = 16$. Now,

$$\begin{aligned} & (0x0x000x0x0x000x)_2 \equiv 0 \pmod{2^{16} - 1} \\ \Longrightarrow\ & x(01010001\|01010001)_2 \equiv 0 \pmod{2^{16} - 1} \\ \Longrightarrow\ & (2^u + 2^v - 3)(2^8 + 1)(01010001)_2 \equiv 0 \pmod{2^{16} - 1} \\ \Longrightarrow\ & (2^u + 2^v - 3)(01010001)_2 \equiv 0 \pmod{2^8 - 1} \\ \Longrightarrow\ & (2^u + 2^v) \cdot 81 \equiv (2^1 + 2^0) \cdot 81 \pmod{2^8 - 1}. \end{aligned}$$

Note that $(01010001)_2 = 81$ is 8-CNA, and $\overset{\hookrightarrow}{\mathrm{ord}}(81) = 8$. Let $S = \{0, 4, 6\}$; then $d(S) = (4, 2, 2)$ and $d(S+4) = (2, 2, 4)$, which is the lexicographically smallest cyclic shift of $d(S)$. Continuing from our last implication,

$$\begin{aligned} \Longrightarrow\ & (S+u) \cup (S+v) = (S+1) \cup S \\ \Longrightarrow\ & (S+u+4) \cup (S+v+4) = (S+5) \cup (S+4) \\ \Longrightarrow\ & (S+u+4) \cup (S+v+4) = \{1, 3, 5\} \cup \{0, 2, 4\}. \end{aligned}$$

We can assume that $0 \in S+v+4$, and then it must be that $1 \in S+u+4$. If $2 \in S+u+4$, this would contradict the fact that $(2,2,4)$ is the smallest difference sequence of all translates of S. Thus, $2 \in S+v+4$ and then $3 \in S+u+4$. Similarly, $4 \in S+v+4$ and $5 \in S+u+4$. Thus,

$$\begin{aligned} & S+u+4 = S+5 \quad \text{and} \quad S+v+4 = S+4 \\ \Longrightarrow\ & u \equiv 1 \pmod 8 \quad \text{and} \quad v \equiv 0 \pmod 8. \end{aligned}$$

Now, $-x = 2^u + 2^v - 3 = 2(2^{u-1} - 1) + (2^v - 1)$. Since $2^8 - 1 \mid 2^{u-1} - 1$ and $2^8 - 1 \mid 2^v - 1$, we have $2^8 - 1 \mid x$. So, if $\{0,1,x\}$ is not an NADS, then it must be that x is divisible by a Mersenne number.

If we take $u, v \in \{2, 3, 4, 5, 6, 7, 8\}$ with $u \neq v$ and set $x = 3 - (2^u + 2^u)$ then, after eliminating multiples of Mersenne numbers, Theorem 7.2 tells us that each of the values $-17, -37, -65, -157, -257, -269, -317$ makes $\{0,1,x\}$ an NADS.

It may not be immediately clear that there are in fact an infinite number of x values with no Mersenne divisors and $w((3-x)/4) = 2$; however, we can deduce this

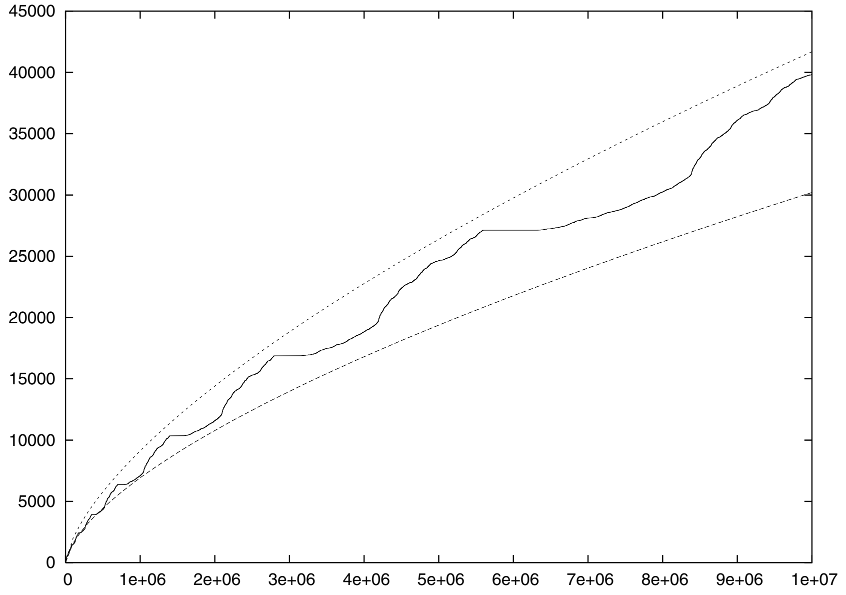


FIG. 8.1. *A plot of* $(-X, c(X))$ *for* $0 \geq X \geq -10^7$.

from Lemma 7.4. If $x = 3 - (2^u + 2^v)$ with $u > v \geq 2$, then, applying Lemma 7.4 with $B = 1$, we see that x has a Mersenne divisor if and only if $\gcd(u, v-1) > 1$ or $\gcd(u-1, v) > 1$. So, we can get an infinite family if we take u's and v's with $\gcd(u, v-1) = \gcd(u-1, v) = 1$. For example, we can take $v = 2$ and then let $u \geq 4$ be any even integer.

Looking at Theorems 7.1 and 7.2, we find that a natural question to ask is if there is an infinite family of NADS with the property that $w(\frac{3-x}{4}) = 3$. One of our results gives a partial answer to this question. If $\frac{3-x}{4} = 11 \cdot 2^i$ with $i \geq 0$, then $w(\frac{3-x}{4}) = 3$; however, Corollary 6.3 tells us that such a value of x will never give an NADS.

8. Further work and comments. It is possible to show that for $n \in \mathbb{Z}^+$ with $n \leq \lfloor -x/3 \rfloor$, the running time of EVAL-$R_D(n)$, as described in Algorithm 4.7, is $O(|x|/3) = O(|x|)$. Thus, to compute EVAL-$R_D(n)$ for all positive integers in this range takes time $O(|x|^2)$. So, we can decide if $\{0, 1, x\}$ is an NADS in $O(|x|^2)$ time. The running time can be reduced to $O(|x|)$ if more memory is used, and this is the approach taken in Algorithm 4.10. However, since the size of the input to this algorithm is $\lg |x|$, the running time is exponential. It would be interesting to determine if there is a polynomial-time algorithm for deciding if $\{0, 1, x\}$ is an NADS.

The "NADS-counting" function is defined as

$$c(X) := |\{x : x \geq X, \{0, 1, x\} \text{ is an NADS}\}| .$$

For example, $c(7) = 0$, $c(3) = 1$, and $c(-1) = 2$. A plot of $c(X)$ for $0 \geq X \geq -10^7$ is given in Figure 8.1 and an interesting fractal structure can be observed. The flat intervals of the plot are the result of Theorem 6.5. The two smooth curves bounding $c(X)$ in Figure 8.1 are $(-X)^{0.64}$ and $(-X)^{0.66}$; these functions were discovered empirically. It would be nice to be able to say something concrete about the asymptotic behavior of $c(X)$.

The function f_D, defined in (4.1), bears some similarity to the Collatz function,

$$f(n) = \begin{cases} n/2 & \text{if } n \text{ is even,} \\ (3n+1)/2 & \text{if } n \text{ is odd.} \end{cases}$$

The Collatz function has received considerable study, but its properties are complex and not well understood. Perhaps this suggests that the study of NADS is also a difficult problem.

Some of our results on NADS appear to have analogues in Matula's [5] theory on basic digit sets. In particular, our Theorem 4.8 corresponds to Matula's Lemma 6, and our Theorem 5.1 corresponds to Matula's Theorem 5. It would be interesting to find other connections between the two works. It might be that our Theorems 7.1 and 7.2, which do not appear to have analogues in [5], may suggest new infinite families of basic digit sets.

Appendix. We list all the values of x from -1 to -10^4 for which $\{0, 1, x\}$ is an NADS:

```
   -1    -5   -13   -17   -25   -29   -37   -53   -61   -65
 -113  -121  -125  -137  -145  -149  -157  -233  -241  -253
 -257  -265  -269  -277  -281  -305  -317  -325  -437  -481
 -485  -493  -505  -509  -517  -521  -533  -541  -557  -565
 -601  -605  -613  -629  -641  -653  -673  -821  -869  -913
 -937  -977  -989 -1013 -1021 -1025 -1033 -1037 -1045 -1061
-1073 -1081 -1097 -1117 -1133 -1145 -1165 -1265 -1273 -1277
-1289 -1297 -1325 -1345 -1349 -1357 -1621 -1637 -1733 -1745
-1765 -1885 -1933 -1949 -1985 -1993 -2017 -2021 -2033 -2041
-2045 -2053 -2069 -2101 -2105 -2113 -2129 -2137 -2141 -2153
-2161 -2165 -2173 -2185 -2189 -2197 -2237 -2273 -2285 -2293
-2297 -2321 -2353 -2365 -2369 -2381 -2393 -2405 -2425 -2497
-2525 -2533 -2557 -2593 -2609 -2621 -2641 -2645 -2669 -2677
-2693 -3245 -3265 -3337 -3385 -3421 -3509 -3541 -3557 -3629
-3653 -3673 -3761 -3797 -3853 -3877 -3881 -3917 -3925 -3929
-3961 -4001 -4033 -4037 -4085 -4093 -4097 -4105 -4117 -4121
-4133 -4141 -4145 -4153 -4157 -4201 -4205 -4217 -4253 -4261
-4273 -4285 -4297 -4337 -4345 -4349 -4373 -4393 -4397 -4469
-4537 -4541 -4573 -4589 -4597 -4601 -4621 -4633 -4645 -4649
-4661 -4693 -4777 -4801 -5021 -5077 -5093 -5101 -5105 -5113
-5129 -5137 -5153 -5165 -5189 -5197 -5213 -5273 -5281 -5365
-5377 -5381 -5393 -5405 -5437 -5441 -6565 -6613 -6773 -6805
-6929 -6973 -7033 -7277 -7333 -7345 -7381 -7393 -7397 -7465
-7477 -7561 -7597 -7613 -7621 -7649 -7741 -7817 -7865 -7877
-7901 -7949 -8045 -8053 -8065 -8069 -8081 -8093 -8101 -8117
-8129 -8165 -8173 -8177 -8185 -8189 -8201 -8213 -8221 -8233
-8237 -8297 -8305 -8317 -8333 -8341 -8369 -8417 -8429 -8437
-8441 -8453 -8485 -8497 -8501 -8573 -8581 -8593 -8597 -8665
-8669 -8681 -8693 -8717 -8725 -8741 -8753 -8789 -8797 -8825
-8837 -8921 -8977 -9089 -9101 -9133 -9157 -9161 -9181 -9209
-9221 -9245 -9341 -9353 -9421 -9425 -9433 -9461 -9473 -9497
-9505 -9509 -9581 -9665 -9673 -9677 -9697 -9761 -9925 -9997
```

Acknowledgments. We thank the referee for bringing [2] to our attention and for a number of other helpful suggestions.

REFERENCES

[1] A. D. Booth, *A signed binary multiplication technique*, Quart. J. Mech. Appl. Math., 4 (1951), pp. 236–240.

[2] E. F. Brickell, D. M. Gordon, K. S. McCurley, and D. B. Wilson, *Fast exponentiation with precomputation (extended abstract)*, in Advances in Cryptology – EUROCRYPT '92, Lecture Notes in Comput. Sci. 658, Springer-Verlag, Berlin, 1993, pp. 200–207. An extended version of the paper is available online from http://research.microsoft.com/∼dbwilson/bgmw/.

[3] D. M. Gordon, *A survey of fast exponentiation methods*, J. Algorithms, 27 (1998), pp. 129–146.

[4] J. Jedwab and C. J. Mitchell, *Minimum weight modified signed-digit representations and fast exponentiation*, Electron. Lett., 25 (1989), pp. 1171–1172.

[5] D. W. Matula, *Basic digit sets for radix representation*, J. Assoc. Comput. Mach., 29 (1982), pp. 1131–1143.

[6] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, FL, 1996.

[7] F. Morain and J. Olivos, *Speeding up the computations on an elliptic curve using addition-subtraction chains*, Theoret. Inform. Appl., 24 (1990), pp. 531–543.

[8] G. W. Reitwiesner, *Binary arithmetic*, in Advances in Computers, Vol. 1, Academic Press, New York, 1960, pp. 231–308.

[9] J. A. Solinas, *Low-Weight Binary Representations for Pairs of Integers*, Tech. report CORR 2001-41, Centre for Applied Cryptographic Research, University of Waterloo, Ontario, 2001. Available online from http://www.cacr.math.uwaterloo.ca/techreports/2001/-corr2001-41.ps.

[10] J. A. Solinas, *An improved algorithm for arithmetic on a family of elliptic curves*, in Advances in Cryptology – CRYPTO '97, Lecture Notes in Comput. Sci. 1294, Springer-Verlag, Berlin, 1997, pp. 357–371. An extended version of the paper is available from http://www.cacr.math.uwaterloo.ca/techreports/1999/corr99-46.ps.

[11] J. A. Solinas, *Efficient arithmetic on Koblitz curves*, Des. Codes Cryptogr., 19 (2000), pp. 195–249.

SIAM J. Discrete Math.
Vol. 19, No. 1, pp. 192–195

AN UPPER BOUND FOR THE d-DIMENSIONAL ANALOGUE OF HEILBRONN'S TRIANGLE PROBLEM[*]

PETER BRASS[†]

Abstract. In this paper it is shown that for any set of n points selected from the d-dimensional unit cube, d odd, the volume of the smallest simplex spanned by the set is $O(n^{-(1+\frac{1}{2d})})$, which is a slight improvement on the only known upper bound $O(n^{-1})$, although still far from the lower bound $\Omega(n^{-d}\log n)$.

Key words. Heilbronn's triangle problem, spanned simplices

AMS subject classifications. 51M16, 52C99

DOI. 10.1137/S0895480103435810

1. Introduction and result.

Heilbronn's triangle problem asks for the smallest value $f^{\text{area\_min}}(n)$ such that any set of n points in the unit square contains three points that determine a triangle of area at most $f^{\text{area\_min}}(n)$. There is a trivial upper bound $f^{\text{area\_min}}(n) = O(\frac{1}{n})$, since one can triangulate any set of n points in the plane, obtaining at least $n-2$ triangles (the minimum if the points are in convex position), which are disjoint and all contained in the unit square, so the smallest of them has area at most $\frac{1}{n-2}$. There is also a simple lower bound of $f^{\text{area\_min}}(n) = \Omega(\frac{1}{n^2})$, which one gets by selecting a set of n lattice points from the $n \times n$ integer lattice square with the property that no three points are collinear. One such set are the points $(t, t^2 \text{mod} n)_{t=1}^{n}$ for n prime; the maximum size of such a set, conjectured to be $2n$, is the well-studied "no-three-in-line problem" [BrMP05, section 10.1]. Since a nondegenerate triangle of integer lattice points has area at least $\frac{1}{2}$, by scaling the set with a factor $\frac{1}{n}$ one obtains n points in the unit square with all triangles of area at least $\frac{1}{2n^2}$. This problem attracted the attention of several famous number theorists [Ro51], [Sch72], [Ro72a], [Ro72b], [Ro73], [Ro76], [KoPS81], who reduced the upper bound in many steps and finally also increased the lower bound [KoPS82], thereby disproving Heilbronn's conjecture (which was $f^{\text{area\_min}}(n) = O(n^{-2})$). The currently best known bounds are

$$c_1 \frac{\log n}{n^2} < f^{\text{area\_min}}(n) < c_2 \frac{e^{c_3\sqrt{\log n}}}{n^{\frac{8}{7}}}$$

from [KoPS81], [KoPS82], for some $c_1, c_2, c_3 > 0$. Also some exact values of $f^{\text{area\_min}}(n)$ were determined [Go72], [CoY02], but apart from a simpler proof of the $\Omega(\frac{\log n}{n^2})$ lower bound [Le00], [BeHL00], and the observation that random choice is quite bad [JiLV02], [GrJ03], no progress on the Heilbronn triangle problem was made since 1982. The higher-dimensional analogue, the smallest $f_d^{\text{vol\_min}}(n)$ such that any set of n points in the d-dimensional unit cube contains $d+1$ points that span a simplex of volume at most $f_d^{\text{vol\_min}}(n)$, was first studied in [Ba01]. There is again a simple lower bound of $\Omega(\frac{1}{n^d})$, which can be obtained, e.g., by choosing n points from the $n \times \cdots \times n$

[*] Received by the editors October 8, 2003; accepted for publication (in revised form) August 31, 2004; published electronically June 22, 2005.

http://www.siam.org/journals/sidma/19-1/43581.html

[†] Department of Computer Science, City College, CUNY, New York, NY 10031 (peter@cs.ccny.cuny.edu).

lattice cube with no $d+1$ in a hyperplane [Ba01]. This was improved in [Le00] to $f_d^{\text{vol\_min}}(n) \geq c\frac{\log n}{n^d}$, which contains the famous lower bound of [KoPS82] as a special case for $d = 2$. But no nontrivial upper bound is known beyond the $O(\frac{1}{n})$ which follows in any dimension by triangulation. Projection methods do not work, for the image of a higher-dimensional simplex is not a triangle; just finding one small triangular face of the simplex is not enough to give a bound on the simplex volume. A bound on the area of the convex hull of $d+1$-tuples would lift to d-dimensional space, but already the area of convex fourgons in the plane seems to behave differently than that of triangles [Sch72]: we do not know any upper bound better than $O(\frac{1}{n})$ for the area of the smallest spanned fourgon, and the lower bound increases to $\Omega(n^{-(1+\frac{1}{k-2})})$ for convex k-gons in the plane [BeHL00].

It is the aim of this note to prove the first nontrivial upper bound on $f_d^{\text{vol\_min}}(n)$.

THEOREM 1. *Any set of n points in the d-dimensional unit cube determines a simplex of volume $O(n^{-(1+\frac{1}{2d})})$ for $d \geq 3$ odd.*

2. Proof of the theorem. Consider a set of n points in the unit cube. The maximum number of points that can be selected from the unit cube with pairwise distance at least $\frac{1}{t}$ is less than $c_1 t^d$. Thus the graph of distances smaller than $\frac{1}{t}$ does not contain an independent set of size $c_1 t^d$, so by Turán's theorem any set of $n \geq c_1 t^d$ points in the unit cube will determine at least $\binom{n}{2} - \text{ex}(n, K_{c_1 t^d}) \geq c_2(\frac{n}{t^d})^2 t^d = c_2 \frac{n^2}{t^d}$ point-pairs with distance less than $\frac{1}{t}$ (where $\text{ex}(n,G)$ denotes the Turán function of the graph G).

These point-pairs determine $c_2\frac{n^2}{t^d}$ directions (which are all distinct, since any coinciding direction pair generates a simplex of volume 0). These directions can be interpreted as points on the unit sphere in d-dimensional space. If we choose around each of these points on the sphere a spherical disc of radius ε, then the sum of the $(d-1)$-dimensional volumes of these discs is $c_3\varepsilon^{d-1}(\frac{n^2}{t^d})$. If this is more than $\binom{d+1}{2}$ times the total area of the sphere, then there is a point on the sphere that belongs to at least $\binom{d+1}{2}$ of the discs. So we can choose from any set of n points for any value of $t \leq (c_1^{-1}n)^{\frac{1}{d}}$ a set of $\binom{d+1}{2}$ point pairs whose directions differ pairwise by less than $c_4(\frac{n^2}{t^d})^{-\frac{1}{d-1}}$. This defines a graph G_1 with $\binom{d+1}{2}$ edges, so it has at least $d+1$ vertices of degree at least one. From this graph G_1 we iteratively remove those edges with both endvertices having degree at least two, until we arrive at a graph G_2 in which each edge has at least one endvertex of degree one. All vertices of degree at least one in G_1 still have degree at least one in G_2. So G_2 is a graph which consists of isolated edges and stars, and has at least $d+1$ vertices. If d is odd, we can select from this graph G_2 a subgraph G_3 of at least $\frac{d+1}{2}$ edges, with $d+1$ vertices. This simplex spanned by the vertices of G_3 is the simplex of small volume claimed in the theorem. Here we need d to be odd; if d is even, we cannot select $d+1$ vertices with at least $\frac{d+1}{2}$ edges, e.g., if the graph G_2 consists only of isolated edges. Selecting one vertex and one edge less and adding an arbitrary point gives a bound which is worse than the trivial bound $O(\frac{1}{n})$.

To bound the volume of the simplex spanned by these $d+1$ points, we partition this point set $V(G_3)$ into $V(G_3) = A \cup B$, where A consists of the centers of the stars and one vertex from each isolated edge, and B contains the remaining vertices. Then A contains at most $\frac{d+1}{2}$ points, and each point in B is joined by an edge to a point in A. We build the simplex and bound the volume incrementally, in three stages:

(1) The set A determines a simplex $\text{conv}(A)$ of dimension $|A|-1 \leq \frac{d-1}{2}$, and the $(|A|-1)$-dimensional volume of this simplex is at most some constant c_5, the

maximum volume of the intersection of the unit cube by any affine subspace of dimension at most $\frac{d-1}{2}$.

(2) Now we add the first of the remaining points; since this new point is joined to the previous simplex by an edge of length at most $\frac{1}{t}$, the volume of the new simplex is at most $\frac{c_5}{t}$.

(3) Finally we add all the other (at least $\frac{d-1}{2}$) remaining points from B. Each of them is joined by an edge of length at most $\frac{1}{t}$ to a point already in the simplex; and this edge has an angle less than $c_4(\frac{n^2}{t^d})^{-\frac{1}{d-1}}$ to a line in the affine hull of that simplex (the edge joining the point added in stage (2) to its neighbor in A). So the distance of each new point to the affine hull of the previous simplex is less than $c_6\frac{1}{t}(\frac{n^2}{t^d})^{-\frac{1}{d-1}}$. Thus after adding the at least $\frac{d-1}{2}$ remaining points, the volume of the simplex is at most

$$c_7\frac{1}{t^{\frac{d+1}{2}}}((\frac{n^2}{t^d})^{-\frac{1}{d-1}})^{\frac{d-1}{2}},$$

which is

$$c_7\frac{1}{t^{\frac{d+1}{2}}}(\frac{n^2}{t^d})^{-\frac{1}{2}} = c_7\frac{1}{n}\frac{t^{\frac{d}{2}}}{t^{\frac{d+1}{2}}} = c_7\frac{1}{t^{\frac{1}{2}}n}.$$

We choose t as large as possible, $t = (c_1^{-1}n)^{\frac{1}{d}}$, and obtain the theorem.

3. Remarks. This bound is of course only a very weak upper bound if one compares it to the $\Omega(\frac{\log n}{n^d})$ lower bound, but we do not have any better. Unfortunately, it also does not work for even dimensions; if we do not have a spanning system of almost parallel edges, which is impossible if $d+1$ is odd, then we lose so much that the trivial upper bound $O(\frac{1}{n})$ is better. It would be interesting to improve the triangulation argument of the trivial upper bound: is there a function $R_d(n) > n^{1+\varepsilon}$ such that each set of n points in d-dimensional space, in general position, has some triangulation using at least $R_d(n)$ simplices [EdPW90]? Then $f_d^{\text{vol\_min}}(n) < \frac{1}{R_d(n)}$. But no nontrivial bound is known for that problem, either.

REFERENCES

[Ba01] G. Barequet, *A lower bound for Heilbronn's triangle problem in d dimensions*, SIAM J. Discrete Math., 14 (2001), pp. 230–236.

[BeHL00] C. Bertram-Kretzberg, T. Hofmeister, and H. Lefmann, *An algorithm for Heilbronn's problem*, SIAM J. Comput., 30 (2000), pp. 383–390.

[BrMP05] P. Brass, W. Moser, and J. Pach, *Research Problems in Discrete Geometry*, Springer-Verlag, 2005, to appear.

[CoY02] F. Comellas and J. L. A. Yebra, *New lower bounds for Heilbronn numbers*, Electron. J. Combin., 9 (2002), Research Paper 6, 10 pp.

[EdPW90] H. Edelsbrunner, F. P. Preparata, and D. B. West, *Tetrahedrizing point sets in three dimensions*, J. Symbolic Comput., 10 (1990), pp. 335–347.

[Go72] M. Goldberg, *Maximizing the smallest triangle made by n points in a square*, Math. Mag., 45 (1972), pp. 135–144.

[GrJ03] G. Grimmett and S. Janson, *On smallest triangles*, Random Structures Algorithms, 23 (2003), pp. 206–223.

[JiLV02] T. Jiang, M. Li, and P. Vitányi, *The average-case area of Heilbronn-type triangles*, Random Structures Algorithms, 20 (2002), pp. 206–219.

[KoPS81] J. Komlós, J. Pintz, and E. Szemerédi, *On Heilbronn's triangle problem*, J. London Math. Soc., 24 (1981), pp. 385–396.

[KoPS82] J. Komlós, J. Pintz, and E. Szemerédi, *A lower bound for Heilbronn's problem*, J. London Math. Soc., 25 (1982), pp. 13–24.

[LeS02] H. Lefmann and N. Schmitt, *A deterministic polynomial-time algorithm for Heilbronn's problem in three dimensions*, SIAM J. Comput., 31 (2002), pp. 1926–1947.

[Le00] H. Lefmann, *On Heilbronn's problem in higher dimension*, in Proceedings of the Eleventh ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, pp. 60–64.

[Ro51] K. F. Roth, *On a problem of Heilbronn*, J. London Math. Soc., 26 (1951), pp. 198–204.

[Ro72a] K. F. Roth, *On a problem of Heilbronn* II, Proc. London Math. Soc., 25 (1972), pp. 193–212.

[Ro72b] K. F. Roth, *On a problem of Heilbronn* III, Proc. London Math. Soc., 25 (1972), pp. 543–549.

[Ro73] K. F. Roth, *Estimation of the area of the smallest triangle obtained by selecting three out of n points in a disc of unit area*, in Analytic Number Theory, H. G. Diamond, ed., Proc. Sympos. Pure Math., 24, AMS, Providence, RI, 1973, pp. 251–262.

[Ro76] K. F. Roth, *Developments in Heilbronn's triangle problem*, Advances in Math., 22 (1976), pp. 364–385.

[Sch72] W. M. Schmidt, *On a problem of Heilbronn*, J. London Math. Soc., 4 (1971/1972), pp. 545–550.

SIAM J. DISCRETE MATH.
Vol. 19, No. 1, pp. 196–207

ENUMERATING TYPICAL CIRCULANT COVERING PROJECTIONS ONTO A CIRCULANT GRAPH*

RONGQUAN FENG†, JIN HO KWAK‡, AND YOUNG SOO KWON‡

Abstract. Enumerating the isomorphism classes of several types of graph covering projections is one of the central research topics in enumerative topological graph theory (see [S. F. Du, D. Marušič, and A. O. Waller, *J. Combin. Theory Ser. B*, 74 (1998), pp. 276–290], [S. F. Du, J. H. Kwak, and M. Y. Xu, *J. Combin. Theory Ser. B*, 93 (2005), pp. 73–93], [R. Feng, J. H. Kwak, J. Kim, and J. Lee, *SIAM J. Discrete Math.*, 11 (1998), pp. 265–272], [R. Feng. and J. H. Kwak, *Discrete Math.*, 277 (2004), pp. 73–85], [C. D. Godsil and A. D. Hensel, *J. Combin. Theory Ser. B.*, 56 (1992), pp. 205–238], [M. Hofmeister, *Discrete Math.*, 143 (1995), pp. 87–97], [M. Hofmeister, *SIAM J. Discrete Math.*, 8 (1995), pp. 51–61], [M. Hofmeister, *SIAM J. Discrete Math.*, 11 (1998), pp. 286–292], [J. H. Kwak, J. Chun, and J. Lee, *SIAM J. Discrete Math.*, 11 (1998), pp. 273–285], [J. H. Kwak and J. Lee, *Canad. J. Math.*, 42 (1990), pp. 747–761], and [J. H. Kwak and J. Lee, *Combinatorial and Computational Mathematics: Present and Future*, (2001), pp. 97–161]). A covering projection is called *circulant* if its covering graph is circulant. A covering projection p from a Cayley graph $\mathrm{Cay}(\mathcal{A}, X)$ onto another $\mathrm{Cay}(\mathcal{Q}, Y)$ is called *typical* if the map $p : \mathcal{A} \to \mathcal{Q}$ on the vertex sets is a group homomorphism from $\mathcal{A}$ onto $\mathcal{Q}$. In [R. Feng. and J. H. Kwak, *Discrete Math.*, 277 (2004), pp. 73–85], the authors enumerated the isomorphism classes of typical circulant double covering projections onto a circulant graph. As a continuation of this work, we enumerate in this paper the isomorphism classes of those covering projections of any folding number.

Key words. graph covering, enumeration, voltage assignment, circulant graph

AMS subject classifications. 05C10, 05C30

DOI. 10.1137/S0895480103431861

1. Introduction.

Throughout this paper, graphs are finite, undirected, simple, and connected. Let G be a graph with vertex set $V(G)$ and edge set $E(G)$. The neighborhood of a vertex $v \in V(G)$, denoted by $N(v)$, is the set of vertices adjacent to v. An *automorphism* of G is a permutation of the vertex set $V(G)$ that preserves adjacency. The set of automorphisms forms a permutation group, called the *automorphism group* $\mathrm{Aut}\,(G)$ of G. A graph G is *vertex-transitive* if $\mathrm{Aut}\,(G)$ acts transitively on the vertex set $V(G)$.

Let $\mathcal{A}$ be a finite group and let X be a subset of $\mathcal{A}$ such that $X = X^{-1}$ (called *symmetric*) and $1 \notin X$. The *Cayley graph* $G = \mathrm{Cay}(\mathcal{A}, X)$ on $\mathcal{A}$ relative to X is the graph having vertex set $V(G) = \mathcal{A}$ and edge set $E(G) = \{\{g, gx\} \,|\, g \in \mathcal{A}, x \in X\}$. For a Cayley graph $G = \mathrm{Cay}(\mathcal{A}, X)$, it is clear that $\mathrm{Aut}\,(G)$ contains the left regular representation $L(\mathcal{A})$ of the group $\mathcal{A}$, and so G is vertex-transitive; and that G is connected if and only if X generates $\mathcal{A}$. The elements of X are called the *connectors* of the graph $\mathrm{Cay}(\mathcal{A}, X)$. A circulant graph of order n is a Cayley graph on the additive group $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$. Circulant graphs are widely applied as models of telecommunication networks (see [1], [14]). It is clear that a circulant graph

*Received by the editors July 21, 2003; accepted for publication (in revised form) October 5, 2004; published electronically July 18, 2005.

http://www.siam.org/journals/sidma/19-1/43186.html

†LMAM, School of Mathematical Sciences, Peking University, Beijing 100871, People's Republic of China (fengrq@math.pku.edu.cn). The work of this author was supported by NSF of China (60473019) and by NKBRPC (2004CB318000).

‡Mathematics, Pohang University of Science and Technology, Pohang, 790-784 Korea (jinkwak@postech.ac.kr, ysookwon@postech.ac.kr). The work of these authors was supported by Com^{2}MaC-KOSEF in Korea.

$\mathrm{Cay}(\mathbb{Z}_n, X)$ is of odd valency if and only if n is even and the order 2 element $\frac{n}{2} \in \mathbb{Z}_n$ is a connector, and that if a circulant graph $\mathrm{Cay}(\mathbb{Z}_n, X)$ with $X = \{\pm i_1, \pm i_2, \ldots, \pm i_k\}$ is connected, then $(i_1, i_2, \ldots, i_k, n) = 1$, where $(i_1, i_2, \ldots, i_k, n)$ denotes the greatest common divisor of $i_1, i_2, \ldots, i_k$ and n. Also, we identify the integers $0, 1, \ldots, n-1$ with their residue classes modulo n.

A *covering projection* (or simply *covering*) from a graph $\widetilde{G}$ to another G is a surjection $p : V(\widetilde{G}) \to V(G)$ such that $p|_{N(\tilde{v})} : N(\tilde{v}) \to N(v)$ is a bijection for all vertices $v \in V(G)$ and $\tilde{v} \in p^{-1}(v)$. Sometimes, a graph $\widetilde{G}$ is also called a covering of G with the projection $p : \widetilde{G} \to G$, and it is ℓ-fold if p is ℓ-to-one. The fibre of an edge or a vertex is its preimage under p. If a covering graph $\widetilde{G}$ is circulant, then p is called a *circulant covering*. A covering projection $p : \widetilde{G} \to G$ is *regular* if there exists a subgroup $\mathcal{B}$ of $\mathrm{Aut}\,(\widetilde{G})$ which acts freely on $\widetilde{G}$, and an isomorphism $i : \widetilde{G}/\mathcal{B} \to G$ such that $i \circ q_{\mathcal{B}} = p$, where $q_{\mathcal{B}} : \widetilde{G} \to \widetilde{G}/\mathcal{B}$ is the quotient map. In this case, the graph $\widetilde{G}$ is called a *regular* covering of the graph G. It is clear that every double covering is regular.

Two coverings, $p_i : \widetilde{G}_i \to G$, $i = 1, 2$, are said to be *isomorphic* if there exists a graph isomorphism $\Phi : \widetilde{G}_1 \to \widetilde{G}_2$ such that $p_2 \circ \Phi = p_1$. Such a Φ is called a *covering isomorphism*.

Every edge of a graph G gives rise to a pair of oppositely directed edges. By $e^{-1} = vu$, we mean the reverse directed edge to a directed edge $e = uv$. A directed edge is also called an *arc* and the set of arcs of the graph G is denoted by $D(G)$. Let $\mathcal{A}$ be a finite group. Following Gross and Tucker [7], a (*ordinary*) *voltage assignment* ϕ of G is a function $\phi : D(G) \to \mathcal{A}$ with the property that $\phi(e^{-1}) = \phi(e)^{-1}$ for each $e \in D(G)$. The *derived graph* G^{ϕ} from a voltage assignment ϕ is defined as follows: $V(G^{\phi}) = V(G) \times \mathcal{A}$, and for each arc $e = uv \in D(G)$ and $a \in \mathcal{A}$, let there be an arc (e, a) in $D(G^{\phi})$ joining vertices (u, a) and $(v, a\phi(e))$. The first coordinate projection $p_{\phi} : G^{\phi} \to G$ is a regular covering.

Let $C^1(G; \mathcal{A})$ denote the set of all voltage assignments $\phi : D(G) \to \mathcal{A}$ of G. Gross and Tucker [7] showed that every regular ℓ-fold covering $\widetilde{G}$ of a graph G can be derived from a voltage assignment in $C^1(G; \mathcal{A})$ for some finite group $\mathcal{A}$ of order ℓ.

A composition of two regular covering projections is not necessarily regular. Although necessary and sufficient conditions for regularity of a composition of two regular coverings have been known by J. Siagiova in [15], [16], they are not easy to apply. Since a Cayley graph can be described as a regular covering of a bouquet of circles, to determine whether a regular covering of a regular covering of a bouquet of circles G is a regular covering of G is equivalent to determine whether a regular covering of a Cayley graph is Cayley. This problem can be generalized by asking which regular coverings of a graph G have a property $\mathcal{P}$ when the base graph G has the property $\mathcal{P}$. For example, Godsil and Hensel [6] considered distance-regularity and Du et al. [2], [3] dealt with 2-arc-transitivity in place of $\mathcal{P}$; due to the overall difficulty of the problem they restricted the base graph to a complete graph. In this paper, we take for $\mathcal{P}$ the property of being circulant. Clearly, a circulant graph can have a noncirculant covering, and a noncirculant graph can have a circulant covering (see [5]). However, it was shown in [5] that no double covering of a circulant graph of valency 3 is circulant.

This paper is organized as follows. In section 2, we review a typical covering of a Cayley graph which was introduced by Feng and Kwak [5]. In section 3, the isomorphism classes of typical circulant prime-fold covering projections onto a circulant graph are enumerated. In section 4, it will be shown that for any composite number $\ell = \ell_1\ell_2$, the number of the isomorphism classes of typical circulant ℓ-fold covering projections onto a circulant graph G is the product of the number of the isomorphism classes of typical circulant ℓ_1-fold covering projections onto G and the number of the isomorphism classes of typical circulant ℓ_2-fold covering projections onto a circulant graph $\widetilde{G}$, where $\widetilde{G}$ is any typical circulant ℓ_1-fold covering of G. As a result, the isomorphism classes of typical circulant ℓ-fold covering projections onto a circulant graph can be enumerated for any natural number ℓ. Its application to a complete graph is also considered. In section 5, it is proved that for any trivalent circulant graph G which is neither K_4 nor $K_{3,3}$, every circulant covering of G is typical. So the isomorphism classes of any finite-fold circulant connected coverings of G are completely enumerated.

2. Typical coverings of a Cayley graph. Let $1 \to \mathcal{K} \to \mathcal{A} \to \mathcal{Q} \to 1$ be a short exact sequence of finite groups with an epimorphism $f : \mathcal{A} \to \mathcal{Q}$. In the following, we identify the group $\mathcal{K}$ with the kernel of the epimorphism f. Choose a symmetric generating set X of the group $\mathcal{A}$ with $1 \notin X$ and let $\mathrm{Cay}(\mathcal{A}, X)$ be the corresponding Cayley graph. As a subgroup of $\mathcal{A}$, the group $\mathcal{K}$ acts freely on the Cayley graph $\mathrm{Cay}(\mathcal{A}, X)$ (by left multiplication), and the quotient projection $q_{\mathcal{K}} : \mathrm{Cay}(\mathcal{A}, X) \to \mathrm{Cay}(\mathcal{A}, X)/\mathcal{K}$ is a regular covering with covering transformation group $\mathcal{K}$. Let $Y = f(X)$. Then Y is a symmetric generating set for the group $\mathcal{Q}$. Furthermore, f induces a covering projection $f_* : \mathrm{Cay}(\mathcal{A}, X) \to \mathrm{Cay}(\mathcal{Q}, Y)$. It is easy to see that the two graph coverings $q_{\mathcal{K}} : \mathrm{Cay}(\mathcal{A}, X) \to \mathrm{Cay}(\mathcal{A}, X)/\mathcal{K}$ and $f_* : \mathrm{Cay}(\mathcal{A}, X) \to \mathrm{Cay}(\mathcal{Q}, Y)$ can be identified through a graph isomorphism $f_\# : \mathrm{Cay}(\mathcal{A}, X)/\mathcal{K} \to \mathrm{Cay}(\mathcal{Q}, Y)$ defined by $a\mathcal{K} \mapsto f_*(a)$. In other words, $f_* = f_\# \circ q_{\mathcal{K}}$. So f_* is a regular covering. Such a covering $f_* : \mathrm{Cay}(\mathcal{A}, X) \to \mathrm{Cay}(\mathcal{Q}, Y)$ is called a *typical* covering derived from an epimorphism f. That is, a covering $p : \mathrm{Cay}(\mathcal{A}, X) \to \mathrm{Cay}(\mathcal{Q}, Y)$ is a typical one derived from an epimorphism $f : \mathcal{A} \to \mathcal{Q}$ if the projection $p : \mathcal{A} \to \mathcal{Q}$ on the vertex sets is the same as the epimorphism f. Thus $p(1_{\mathcal{A}}) = 1_{\mathcal{Q}}$ and then $p(X) = Y$. Note that in a typical covering $f_* : \mathrm{Cay}(\mathcal{A}, X) \to \mathrm{Cay}(\mathcal{Q}, Y)$ derived from an epimorphism f, if there is an $x \in X \cap \mathcal{K}$, or there exist two distinct elements x and x' in X such that $f(x) = f(x')$, then the graph $\mathrm{Cay}(\mathcal{Q}, Y)$ cannot be simple. Therefore, we assume that $X \cap \mathcal{K} = \emptyset$ and $f(x) \neq f(x')$ for any $x \neq x'$ in X in order to deal with only simple graphs throughout this paper.

It is shown in [5] that a circulant covering of a circulant graph is not necessarily a typical one.

3. Typical circulant coverings of a prime-fold. Let $G = \mathrm{Cay}(\mathbb{Z}_n, Y)$ be a connected circulant graph on the group $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$ and let ℓ be any natural number. In order to enumerate the isomorphism classes of typical circulant ℓ-fold coverings of the graph G, let us consider a short exact sequence $0 \to \mathbb{Z}_\ell \to \mathbb{Z}_{\ell n} \to \mathbb{Z}_n \to 0$ with an epimorphism $f : \mathbb{Z}_{\ell n} \to \mathbb{Z}_n$. Then $\alpha := f(1)$ must be a generator of the cyclic group $\mathbb{Z}_n$, so that $(\alpha, n) = 1$. Denote by f_α the epimorphism $f : \mathbb{Z}_{\ell n} \to \mathbb{Z}_n$ such that $f(1) = \alpha$. Then, for any $a \in \mathbb{Z}_{\ell n}$, we have $f_\alpha(a) = \alpha a \pmod{n}$. Therefore, the kernel of f_α is $\{0, n, \ldots, (\ell-1)n\}$. In what follows, let $\mathcal{K} = \{0, n, \ldots, (\ell-1)n\}$.

For a regular covering, all connected components of a covering graph are isomorphic as covering graphs. So it is enough to enumerate connected ones. In this section, let $\ell = p$ be a prime. The case of $p = 2$ has been studied already in [5].

THEOREM 1 (see [5]). *Let G be a connected circulant graph of order n. If the valency of G is odd, there is no typical circulant double covering of G. If the valency of G is even, say $2k$, then the number of isomorphism classes of typical circulant connected double coverings of G is $2^k - 1$ if n is odd, and 2^{k-1} if n is even.*

Throughout this section, let p be an odd prime if not stated otherwise.

Lemma 2. *Any typical covering* $f_* : \mathrm{Cay}(\mathbb{Z}_{pn}, X) \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ *can be derived from an epimorphism* f *satisfying* $f(1) = 1$. *That is, any typical covering* $f_{\alpha *} : \mathrm{Cay}(\mathbb{Z}_{pn}, X) \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ *derived from the epimorphism* f_α *is isomorphic to a typical covering* $f_{1*} : \mathrm{Cay}(\mathbb{Z}_{pn}, X') \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ *derived from the epimorphism* f_1.

Proof. Let $f_* : \mathrm{Cay}(\mathbb{Z}_{pn}, X) \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ be any typical covering derived from an epimorphism $f : \mathbb{Z}_{pn} \to \mathbb{Z}_n$ and let $f(1) = \alpha$. First, let $(\alpha, p) = 1$. Then $(\alpha, pn) = 1$ because $(\alpha, n) = 1$. Define $\Phi_\alpha(a) = \alpha a$ for any $a \in \mathbb{Z}_{pn}$. The map Φ_α is clearly an automorphism of the cyclic group $\mathbb{Z}_{pn}$. Set $X' = \Phi_\alpha(X)$. Now, Φ_α induces a graph isomorphism from $\mathrm{Cay}(\mathbb{Z}_{pn}, X)$ to $\mathrm{Cay}(\mathbb{Z}_{pn}, X')$. Let $f_1 : \mathbb{Z}_{pn} \to \mathbb{Z}_n$ denote the epimorphism defined by $f_1(1) = 1$ as before. Then, we have $f_1\Phi_\alpha(a) = f_1(\alpha a) = \alpha a = f(a)$ for any $a \in \mathbb{Z}_{pn}$, which implies $f_1 \circ \Phi_\alpha = f$. Therefore, the two coverings f_* and f_{1*} are isomorphic via the covering isomorphism Φ_α. Furthermore, since $f_1(X') = f_1(\Phi_\alpha(X)) = f(X) = Y$, $X' \cap \{0, n, \ldots, (p-1)n\} = \emptyset$ and $f_1(x) \neq f_1(x')$ for any $x \neq x'$ in X', the covering $f_{1*} : \mathrm{Cay}(\mathbb{Z}_{pn}, X') \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ is also a typical one.

If $(\alpha, p) \neq 1$, or equivalently, $p|\alpha$, then $(p, n) = 1$ because $(\alpha, n) = 1$. Let $\alpha' = \alpha + n$, then $(\alpha', pn) = 1$. Define $\Phi_{\alpha'} : \mathbb{Z}_{pn} \to \mathbb{Z}_{pn}$ by $\Phi_{\alpha'}(a) = \alpha' a$ for any $a \in \mathbb{Z}_{pn}$. Then $\Phi_{\alpha'}$ is an automorphism of the cyclic group $\mathbb{Z}_{pn}$. Set $X' = \Phi_{\alpha'}(X)$. By repeating the same process as the previous case, one can show that the covering $f_* : \mathrm{Cay}(\mathbb{Z}_{pn}, X) \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ is isomorphic to the typical covering $f_{1*} : \mathrm{Cay}(\mathbb{Z}_{pn}, X') \to \mathrm{Cay}(\mathbb{Z}_n, Y)$. □

Following Lemma 2, one may assume that every typical circulant p-fold covering of a circulant graph $G = \mathrm{Cay}(\mathbb{Z}_n, Y)$ is derived from the epimorphism $f : \mathbb{Z}_{pn} \to \mathbb{Z}_n$ defined by $f(1) = 1$ throughout this section. Therefore, every typical circulant p-fold covering $f_* : \mathrm{Cay}(\mathbb{Z}_{pn}, X) \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ can be obtained by choosing a generating set X of $\mathbb{Z}_{pn}$ so that $f(X) = Y$ and $f|_X : X \to Y$ is injective. In what follows, we define a voltage assignment to derive a typical circulant p-fold covering.

Let $G = \mathrm{Cay}(\mathbb{Z}_n, Y)$ be a circulant graph. Assume that $Y = \{\pm i_1, \pm i_2, \ldots, \pm i_k\}$ or $\{\pm i_1, \pm i_2, \ldots, \pm i_k, \frac{n}{2}\}$ according as the valency of G is even $2k$ or odd $2k+1$, where $0 < i_1, i_2, \ldots, i_k < \left\lfloor \frac{n+1}{2} \right\rfloor$. To construct a typical circulant p-fold covering of G, we define a voltage assignment $\phi : D(G) \to \mathbb{Z}_p = \{0, 1, \ldots, p-1\}$ as follows: Let $\delta = (\delta_1, \delta_2, \ldots, \delta_k) \in \mathbb{Z}_p^k$ be any k-tuple. For any arc $e = (a, a + i_\alpha)$ in $D(G)$ with a connector i_α, define

$$\phi(e) = \begin{cases} \delta_\alpha & \text{if } a < a + i_\alpha, \\ \delta_\alpha + 1 & \text{if } a > a + i_\alpha, \end{cases} \tag{1}$$

and $\phi(e^{-1}) = -\phi(e)$. If the valency of G is odd $2k+1$, then we define in addition

$$\phi(e) = \begin{cases} \frac{1}{2}(p-1) & \text{if } a < a + \frac{n}{2}, \\ \frac{1}{2}(p-1) + 1 & \text{if } a > a + \frac{n}{2}, \end{cases} \tag{2}$$

for any arc $e = (a, a + \frac{n}{2})$ determined by the connector $\frac{n}{2}$. In this case, we say that ϕ is of type $(\delta_1, \delta_2, \ldots, \delta_k)$. Such a voltage assignment ϕ defined by the (1) and (2) is called *typical*.

Now assume that ϕ is a typical voltage assignment of type $(\delta_1, \delta_2, \ldots, \delta_k)$. Set $X = \{\pm j_1, \pm j_2, \ldots, \pm j_k\}$ or $X = \{\pm j_1, \pm j_2, \ldots, \pm j_k, \frac{pn}{2}\}$ depending on $Y = \{\pm i_1, \pm i_2, \ldots, \pm i_k\}$ or $\{\pm i_1, \pm i_2, \ldots, \pm i_k, \frac{n}{2}\}$, respectively, where $j_\alpha = i_\alpha + \delta_\alpha n$, $1 \leq \alpha \leq k$. It is clear that $f(X) = Y$ and $\mathrm{Cay}(\mathbb{Z}_{pn}, X)$ is a typical circulant p-fold covering of the graph G. Define $\Phi : V(G^{\phi}) \to \mathbb{Z}_{pn}$ by $\Phi(a,m) = a + mn$, where $a \in V(G) = \mathbb{Z}_n = \{0,1,\ldots,n-1\}$ and $m \in \mathbb{Z}_p = \{0,1,\ldots,p-1\}$. It is clear that Φ is a bijection. Moreover, if two vertices (a_1, m_1) and (a_2, m_2) are adjacent in the graph G^{ϕ}, then a_1 and a_2 are adjacent in G and $m_2 = m_1 + \phi(a_1a_2)$. Let the arc a_1a_2 be determined by some connector $i_\alpha \in Y$. Then, as elements in $\mathbb{Z}_{pn}$,

$$a_2 - a_1 = \begin{cases} i_\alpha & \text{if } a_1 < a_2, \\ -n + i_\alpha & \text{if } a_1 > a_2 \end{cases}$$

$$\text{and } \mathrm{m}_2 - \mathrm{m}_1 = \phi(\mathrm{a}_1\mathrm{a}_2) = \begin{cases} \delta_\alpha & \text{if } a_1 < a_2, \\ \delta_\alpha + 1 & \text{if } a_1 > a_2. \end{cases}$$

So $\Phi(a_2,m_2) - \Phi(a_1,m_1) = (a_2 - a_1) + (m_2 - m_1)n = i_\alpha + \delta_\alpha n = j_\alpha$. Thus the two vertices $\Phi(a_1,m_1)$ and $\Phi(a_2,m_2)$ are adjacent in the graph $\mathrm{Cay}(\mathbb{Z}_{pn}, X)$. Moreover, the two graphs G^{ϕ} and $\mathrm{Cay}(\mathbb{Z}_{pn}, X)$ have the same number of edges. Thus Φ is a graph isomorphism from G^{ϕ} to $\mathrm{Cay}(\mathbb{Z}_{pn}, X)$. Furthermore, we have $f_* \circ \Phi(a,m) = f(a + mn) = a = p_\phi(a,m)$ for any $(a,m) \in V(G^{\phi})$, implying $f_* \circ \Phi = p_\phi$. Therefore, the two covering projections p_ϕ and f_* are isomorphic through the covering isomorphism Φ.

Conversely, if a typical circulant p-fold covering $f_* : \mathrm{Cay}(\mathbb{Z}_{pn}, X) \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ is given with $X = \{\pm j_1, \pm j_2, \ldots, \pm j_k\}$ or $X = \{\pm j_1, \pm j_2, \ldots, \pm j_k, \frac{pn}{2}\}$, then $f(X) = Y = \{\pm i_1, \pm i_2, \ldots, \pm i_k\}$ or $f(X) = \{\pm i_1, \pm i_2, \ldots, \pm i_k, \frac{n}{2}\}$. Thus, for any $\alpha = 1, 2, \ldots, k$, $j_\alpha - i_\alpha \in \mathcal{K}$ and then $j_\alpha = i_\alpha + \delta_\alpha n$ for some $\delta_\alpha \in \mathbb{Z}_p$ with $0 < i_\ell < \lfloor \frac{n+1}{2} \rfloor$ for suitable reordering of the subscripts. By chasing the reverse direction of the previous paragraph, one can show that f_* is isomorphic to the covering projection p_ϕ, where ϕ is a typical voltage assignment of type $(\delta_1, \delta_2, \ldots, \delta_k)$. So far, we have proved that a covering projection onto the circulant graph $G = \mathrm{Cay}(\mathbb{Z}_n, Y)$ is a typical circulant p-fold one if and only if it is isomorphic to one of the covering projection p_ϕ, where ϕ is a typical voltage assignment of type $(\delta_1, \delta_2, \ldots, \delta_k)$ for some $\delta = (\delta_1, \delta_2, \ldots, \delta_k) \in \mathbb{Z}_p^k$. This gives the proof of the first part of the following lemma.

Lemma 3. *Any typical circulant p-fold covering of a circulant graph G can be derived from a typical voltage assignment. Furthermore, there exist exactly* $p^{\lfloor \frac{d}{2} \rfloor}$ *typical voltage assignments in* $C^1(G; \mathbb{Z}_p)$*, where d is the valency of G.*

Proof. From the definition of a typical voltage assignment, the voltages of the arcs determined by any fixed connector i_α are completely determined by $\phi(0, i_\alpha)$. Since $\phi(0, i_\alpha) \in \mathbb{Z}_p$ has p choices and there are $\lfloor \frac{d}{2} \rfloor$ such connectors i_α, $1 \le \alpha \le \lfloor \frac{d}{2} \rfloor$, there exist $p^{\lfloor \frac{d}{2} \rfloor}$ typical voltage assignments in $C^1(G; \mathbb{Z}_p)$. □

Kwak and Lee [12] obtained an algebraic characterization of two coverings of a graph G to be isomorphic. The following lemma is a special case of their characterization.

Lemma 4. *Let* ϕ *and* ψ *be typical voltage assignments in* $C^1(G; \mathbb{Z}_p)$*. Then, two typical circulant p-fold coverings* $p_\phi : G^{\phi} \to G$ *and* $p_\psi : G^{\psi} \to G$ *are isomorphic if and only if there exists a function* $g : V(G) \to \mathbb{Z}_p$ *such that* $\psi(uv) = -g(u) + \phi(uv) + g(v)$ *for each* $uv \in D(G)$*.*

Now, let p divide n and let ϕ be a typical voltage assignment of $G = \mathrm{Cay}(\mathbb{Z}_n, Y)$. Since the graph G is assumed to be connected, at least one generator in Y is not divisible by p. Without any loss of generality, one may assume that i_1 is not divisible by p. For each $\lambda \in \mathbb{Z}_p$, define a new voltage assignment $\phi_\lambda : D(G) \to \mathbb{Z}_p$ by $\phi_\lambda(e) =$

$\phi(e) + i_1^{-1} i_\alpha \lambda$ for any arc $e = (a, a + i_\alpha)$ determined by i_α and $\phi_\lambda(e^{-1}) = -\phi_\lambda(e)$. Such an assignment ϕ_λ is well-defined because $(i_1, p) = 1$. Clearly, $\phi_0 = \phi$ and ϕ_λ is also a typical voltage assignment in $C^1(G; \mathbb{Z}_p)$. Define $g : V(G) \to \mathbb{Z}_p$ by $g(u) = i_1^{-1} \lambda u \pmod{p}$. Then, $\phi_\lambda(uv) = -g(u) + \phi(uv) + g(v)$ for any $uv \in D(G)$. So, the two coverings p_{ϕ_λ} and p_ϕ are isomorphic by Lemma 4.

LEMMA 5. *Let $G = \mathrm{Cay}(\mathbb{Z}_n, Y)$ be a circulant graph, and let ϕ, ψ be any two typical voltage assignments in $C^1(G; \mathbb{Z}_p)$. Then, two coverings p_ϕ and p_ψ are isomorphic if and only if*

(1) *$\psi = \phi$ when $(p, n) = 1$, and*

(2) *$\psi = \phi_\lambda$ for some $\lambda \in \mathbb{Z}_p$ when $(p, n) = p$.*

Proof. The sufficiency is already shown. For necessity, suppose that p_ϕ and p_ψ are isomorphic, but $\phi \neq \psi$. Then, there exists an arc u_0v_0 such that $\psi(u_0v_0) \neq \phi(u_0v_0)$. Set $\psi(u_0v_0) - \phi(u_0v_0) = \xi$. Let the arc u_0v_0 be determined by a connector i_α. Then, for any other arc uv determined by the same connector i_α, $\psi(uv) - \phi(uv) = \xi$ because both ϕ and ψ are typical. So, $g(v) - g(u) = \xi$ for the function g in Lemma 4.

Case 1. Let $(p, n) = 1$. The length of every cycle of G determined by the connector i_α is not divided by p (in fact, its length is $n/(n, i_\alpha)$). If a sequence of vertices $u_0, u_1, \ldots, u_{s-1}, u_0$ is one of such cycles of length $s = n/(n, i_\alpha)$, then

$$\begin{array}{rcl} g(u_1) - g(u_0) & = & \xi, \\ g(u_2) - g(u_1) & = & \xi, \\ & \vdots & \\ g(u_0) - g(u_{s-1}) & = & \xi. \end{array}$$

Adding these s equations together, one can get that $0 = s\xi$ in $\mathbb{Z}_p$, which is impossible since $(s, p) = 1$ and $\xi \neq 0$. It implies that if $(p, n) = 1$, then two typical p-fold coverings p_ϕ and p_ψ are isomorphic only when $\psi = \phi$.

Case 2. Let $p|n$. Suppose that the connector i_α is divisible by p. Note that the graph G has at least one connector which is not divided by p because $p|n$ and G is connected. Choose a connector i_β such that $(p, i_\beta) = 1$. Set $s_1 = \frac{i_\beta}{(i_\alpha, i_\beta)}$ and $t_1 = \frac{i_\alpha}{(i_\alpha, i_\beta)}$. Then, $(p, s_1) = 1$ and $p|t_1$ because p does not divide (i_α, i_β). Now, we have a sequence of vertices in the graph G,

$$0,\ i_\alpha,\ 2i_\alpha,\ \ldots,\ (s_1 - 1)i_\alpha,\ s_1 i_\alpha = t_1 i_\beta,\ (t_1 - 1)i_\beta,\ \ldots,\ 2i_\beta,\ i_\beta,\ 0.$$

Therefore,

$$g(i_\alpha) - g(0) = g(2i_\alpha) - g(i_\alpha) = \cdots = g(s_1 i_\alpha) - g((s_1 - 1)i_\alpha) = \xi,$$

which implies $g(s_1 i_\alpha) - g(0) = s_1 \xi \neq 0$, and

$$g(i_\beta) - g(0) = g(2i_\beta) - g(i_\beta) = \cdots = g(t_1 i_\beta) - g((t_1 - 1)i_\beta) = \psi((0, i_\beta)) - \phi((0, i_\beta)),$$

which implies $g(t_1 i_\beta) - g(0) = t_1(\psi((0, i_\beta)) - \phi((0, i_\beta))) = 0$, a contradiction. Therefore, the connector i_α cannot be divisible by p.

Without any loss of generality, one may assume that $i_\alpha = i_1$. For any other connector $i_\gamma \in Y$, set $s_2 = \frac{i_\gamma}{(i_1, i_\gamma)}$ and $t_2 = \frac{i_1}{(i_1, i_\gamma)}$. The same procedure as above to the sequence of vertices

$$0,\ i_1,\ 2i_1,\ \ldots,\ (s_2 - 1)i_1,\ s_2 i_1 = t_2 i_\gamma,\ (t_2 - 1)i_\gamma,\ \ldots,\ 2i_\gamma,\ i_\gamma,\ 0$$

gives $s_2\xi = g(s_2 i_1) - g(0) = g(t_2 i_\gamma) - g(0) = t_2(\psi((0,i_\gamma)) - \phi((0,i_\gamma)))$, which implies $\psi((0,i_\gamma)) - \phi((0,i_\gamma)) = i_1^{-1} i_\gamma \xi$. Therefore, $\psi = \phi_\xi$. □

From Lemmas 3 and 5, one can show easily the following theorem.

THEOREM 6. *Let $G = \mathrm{Cay}(\mathbb{Z}_n, Y)$ be a connected circulant graph of order n. Then, for any odd prime p, the number of isomorphism classes of typical circulant (connected or not) p-fold coverings of G is $p^{\lfloor \frac{d}{2} \rfloor}$ if $(p,n) = 1$, and is $p^{\lfloor \frac{d}{2} \rfloor - 1}$ if $p|n$, where d is the valency of G.*

The following lemma shows how one can enumerate the isomorphism classes of typical circulant connected p-fold coverings of G.

LEMMA 7. *Let $G = \mathrm{Cay}(\mathbb{Z}_n, Y)$ be a connected circulant graph and let p be an odd prime.*

(1) *If $p|n$, then every typical circulant p-fold covering of G is connected, and*

(2) *if $(p,n) = 1$, then there exists one and only one disconnected typical circulant p-fold covering of G.*

Proof. Let $\widetilde{G} = \mathrm{Cay}(\mathbb{Z}_{pn}, X)$ be a typical circulant p-fold covering of G. Write $X = \{\pm j_1, \pm j_2, \ldots, \pm j_k\}$ or $\{\pm j_1, \pm j_2, \ldots, \pm j_k, \frac{pn}{2}\}$ according to $Y = \{\pm i_1, \pm i_2, \ldots, \pm i_k\}$ or $\{\pm i_1, \pm i_2, \ldots, \pm i_k, \frac{n}{2}\}$ and let $\mu = (pn, j_1, j_2, \ldots, j_k)$.

(1) Let $p|n$. Since G is connected, $(i_1, i_2, \ldots, i_k, n) = 1$ and so at least one connector of $i_1, i_2, \ldots, i_k$ is not divisible by p, say i_1. Since $j_1 = i_1 + \delta_1 n$ for some $\delta_1 \in \mathbb{Z}_p$, $(p, j_1) = 1$ and then $(p,\mu) = 1$. Because $\mu|pn$ and $\mu|j_\alpha$ for all $1 \le \alpha \le k$, we have $\mu|n$ and $\mu|i_\alpha$ for all $1 \le \alpha \le k$. It implies that $\mu = 1$ and $\widetilde{G}$ is connected.

(2) Let $(p,n) = 1$. For any integer i, exactly one integer from the set $S = \{i, i+n, \ldots, i+(p-1)n\}$ is divisible by p. So, for any connector i_α, $1 \le \alpha \le k$, we can choose the unique $\delta_\alpha \in \mathbb{Z}_p$ such that $p|(i_\alpha + \delta_\alpha n)$. Let a typical voltage assignment ϕ be defined to be of type $(\delta_1, \delta_2, \ldots, \delta_k)$. Then, G^ϕ is disconnected. Conversely, let $\widetilde{G}$ be a disconnected typical circulant p-fold covering of G. If $(p,\mu) = 1$, then one can prove that $\mu = 1$ as in case (1), a contradiction. Consequently, $p|\mu$ and every j_α $(1 \le \alpha \le k)$ must be divided by p. That is, $j_\alpha = i_\alpha + \delta_\alpha n$, where δ_α is the unique element in $\mathbb{Z}_p$ such that $p|(i_\alpha + \delta_\alpha n)$. Therefore, $\widetilde{G}$ is the same as the covering graph G^ϕ constructed above. □

Now, by Theorem 6 and Lemma 7, we have the following theorem.

THEOREM 8. *Let $G = \mathrm{Cay}(\mathbb{Z}_n, Y)$ be a connected circulant graph of order n. Then, for any odd prime p, the number of isomorphism classes of typical circulant connected p-fold coverings of G is $p^{\lfloor \frac{d}{2} \rfloor} - 1$ if $(p,n) = 1$, and is $p^{\lfloor \frac{d}{2} \rfloor - 1}$ otherwise, where d is the valency of G.*

Note that the above theorem does not hold for even prime $p = 2$. In particular, there are no typical circulant double coverings of a circulant graph G if its valency d is odd. (See Theorem 1.)

4. Typical circulant coverings of any finite-fold. In this section, we enumerate the isomorphism classes of typical circulant ℓ-fold coverings for any positive integer ℓ. All coverings in this section are assumed to be connected ones.

LEMMA 9. *The composition of any two typical circulant covering projections is typical. Conversely, for a composite number $\ell = \ell_1 \ell_2$, any typical circulant ℓ-fold covering projection is a composition of a typical circulant ℓ_1-fold covering projection and a typical circulant ℓ_2-fold covering projection.*

Proof. The first statement is clear. Now suppose that $f_* : \mathrm{Cay}(\mathbb{Z}_{\ell n}, X) \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ is a typical circulant covering derived from an epimorphism $f : \mathbb{Z}_{\ell n} \to \mathbb{Z}_n$. Let $f(1) = \alpha$. Then $(\alpha, n) = 1$ and $f(a) = \alpha a \pmod{n}$. Define two homomorphisms $f_1 : \mathbb{Z}_{\ell n} \to \mathbb{Z}_{\ell_2 n}$ and $f_2 : \mathbb{Z}_{\ell_2 n} \to \mathbb{Z}_n$ by $f_1(1) = 1$ and $f_2(1) = \alpha$, respectively, *i.e.*,

$f_1(a) = a \pmod{\ell_2 n}$ and $f_2(b) = \alpha b \pmod{n}$ for any $a \in \mathbb{Z}_{\ell n}$ and any $b \in \mathbb{Z}_{\ell_2 n}$. Then, $f = f_2 \circ f_1$ and both f_1 and f_2 are epimorphisms. Let $Z = f_1(X)$, then it is easy to check that $f_{1*} : \mathrm{Cay}(\mathbb{Z}_{\ell n}, X) \to \mathrm{Cay}(\mathbb{Z}_{\ell_2 n}, Z)$ is a typical circulant ℓ_1-fold covering derived from the epimorphism f_1 and $f_{2*} : \mathrm{Cay}(\mathbb{Z}_{\ell_2 n}, Z) \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ is a typical circulant ℓ_2-fold covering derived from the epimorphism f_2, and that f_* is the composition of f_{1*} and f_{2*}. □

LEMMA 10. *Let $f_* : \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_1) \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ and $h_* : \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_2) \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ be two isomorphic typical circulant ℓ-fold coverings of the same graph. Then, there is a graph isomorphism $\Phi : \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_1) \to \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_2)$ such that $h_* \circ \Phi = f_*$ and $\Phi(X_1) = X_2$. Moreover, the restriction of Φ on the vertex set $\mathbb{Z}_{\ell n}$ is an automorphism of the cyclic group $\mathbb{Z}_{\ell n}$.*

Proof. Let two typical circulant coverings f_* and h_* be isomorphic through a covering isomorphism $\Psi : \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_1) \to \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_2)$. Since $h_*(\Psi(0)) = f_*(0) = 0$, $\Psi(0) \in \{0, n, \ldots, (\ell-1)n\}$, the kernel of the epimorphism h. Let $\Psi(0) = tn$. Define $\Phi = R_{-tn} \circ \Psi$, where $R_{-tn} : \mathbb{Z}_{\ell n} \to \mathbb{Z}_{\ell n}$ is defined by $R_{-tn}(a) = a - tn$ for $a \in \mathbb{Z}_{\ell n}$. Then R_{-tn} is an automorphism of the graph $\mathrm{Cay}(\mathbb{Z}_{\ell n}, X_2)$. Therefore, Φ is a graph isomorphism from $\mathrm{Cay}(\mathbb{Z}_{\ell n}, X_1)$ to $\mathrm{Cay}(\mathbb{Z}_{\ell n}, X_2)$ and $\Phi(0) = 0$. For any $a \in \mathbb{Z}_{\ell n}$, we have $h_* \circ \Phi(a) = h_* \circ R_{-tn} \circ \Psi(a) = h_*(\Psi(a) - tn) = h_* \Psi(a) = f_*(a)$. Thus $h_* \circ \Phi = f_*$. Furthermore, $\Phi(N(0)) = N(0)$, which implies $\Phi(X_1) = X_2$. For any two connectors $j_{\alpha_1}, j_{\alpha_2} \in X_1$ (not necessarily distinct), $(\Phi(j_{\alpha_1}), \Phi(j_{\alpha_1} + j_{\alpha_2})) = \Phi((j_{\alpha_1}, j_{\alpha_1} + j_{\alpha_2}))$ is an arc in the graph $\mathrm{Cay}(\mathbb{Z}_{\ell n}, X_2)$. Therefore, there exists a connector $j'_{\alpha_2} \in X_1$ such that $\Phi(j_{\alpha_1} + j_{\alpha_2}) = \Phi(j_{\alpha_1}) + \Phi(j'_{\alpha_2})$. From

$$\begin{aligned} f_*(j_{\alpha_1}) + f_*(j_{\alpha_2}) = f_*(j_{\alpha_1} + j_{\alpha_2}) &= h_*(\Phi(j_{\alpha_1} + j_{\alpha_2})) = h_*(\Phi(j_{\alpha_1}) + \Phi(j'_{\alpha_2})) \\ &= h_* \circ \Phi(j_{\alpha_1}) + h_* \circ \Phi(j'_{\alpha_2}) \\ &= f_*(j_{\alpha_1}) + f_*(j'_{\alpha_2}), \end{aligned}$$

we get $f_*(j_{\alpha_2}) = f_*(j'_{\alpha_2})$ which implies $j_{\alpha_2} = j'_{\alpha_2}$. Thus $\Phi(j_{\alpha_1} + j_{\alpha_2}) = \Phi(j_{\alpha_1}) + \Phi(j_{\alpha_2})$ for any $j_{\alpha_1}, j_{\alpha_2} \in X_1$. Since $\mathbb{Z}_{\ell n}$ is generated by X_1, $\Phi(a + a') = \Phi(a) + \Phi(a')$ for any $a, a' \in \mathbb{Z}_{\ell n}$. Thus, the restriction of Φ on $\mathbb{Z}_{\ell n}$ is an automorphism of $\mathbb{Z}_{\ell n}$. □

In what follows, when two typical circulant ℓ-fold coverings $f_* : \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_1) \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ and $h_* : \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_2) \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ are isomorphic through a covering isomorphism $\Phi : \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_1) \to \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_2)$, we always assume that $\Phi(0) = 0$, $\Phi(X_1) = X_2$ and the restriction of Φ on the vertex set $\mathbb{Z}_{\ell n}$ is an automorphism of the group $\mathbb{Z}_{\ell n}$. For a composite number $\ell = \ell_1 \ell_2$, as in the proof of Lemma 9, we assume that f_* is the composition of typical circulant coverings $f_{1*} : \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_1) \to \mathrm{Cay}(\mathbb{Z}_{\ell_2 n}, Z_1)$ and $f_{2*} : \mathrm{Cay}(\mathbb{Z}_{\ell_2 n}, Z_1) \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ and h_* is the composition of typical circulant coverings $h_{1*} : \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_2) \to \mathrm{Cay}(\mathbb{Z}_{\ell_2 n}, Z_2)$ and $h_{2*} : \mathrm{Cay}(\mathbb{Z}_{\ell_2 n}, Z_2) \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ such that $f_1(1) = 1$, $f_2(1) = f(1)$ and $h_1(1) = 1$, $h_2(1) = h(1)$, where $Z_1 = f_1(X_1)$ and $Z_2 = h_1(X_2)$. Then, $h_{1*} \circ \Phi : \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_1) \to \mathrm{Cay}(\mathbb{Z}_{\ell_2 n}, Z_2)$ is also a typical covering, because $h_1 \circ \Phi : \mathbb{Z}_{\ell n} \to \mathbb{Z}_{\ell_2 n}$ is an epimorphism and $h_1 \circ \Phi(X_1) = h_1(X_2) = Z_2$ (see Figure 1).

LEMMA 11. *For any two typical circulant ℓ-fold coverings $f_* : \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_1) \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ and $h_* : \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_2) \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ of the same graph, they are isomorphic through a covering isomorphism $\Phi : \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_1) \to \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_2)$ if and only if the typical circulant coverings f_{2*} and h_{2*} are isomorphic through some covering isomorphism $\Phi' : \mathrm{Cay}(\mathbb{Z}_{\ell_2 n}, Z_1) \to \mathrm{Cay}(\mathbb{Z}_{\ell_2 n}, Z_2)$, and the typical circulant coverings $\Phi' \circ f_{1*}$ and h_{1*} are isomorphic through the covering isomorphism Φ.*

Proof. If f_{2*} and h_{2*} are isomorphic through a covering isomorphism $\Phi' : \mathrm{Cay}(\mathbb{Z}_{\ell_2 n}, Z_1) \to \mathrm{Cay}(\mathbb{Z}_{\ell_2 n}, Z_2)$, and $\Phi' \circ f_{1*}$ and h_{1*} are isomorphic through

$$\begin{tikzcd}
\mathrm{Cay}(\mathbb{Z}_{\ell n}, X_1) \arrow[rr, "\Phi"] \arrow[d, "f_{1*}"'] \arrow[rrd, "\Phi' \circ f_{1*} = h_{1*} \circ \Phi"] \arrow[rdd, "f_*"] & & \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_2) \arrow[d, "h_{1*}"] \arrow[ldd, "h_*"'] \\
\mathrm{Cay}(\mathbb{Z}_{\ell_2 n}, Z_1) \arrow[rr, "\Phi'"] \arrow[rd, "f_{2*}"'] & & \mathrm{Cay}(\mathbb{Z}_{\ell_2 n}, Z_2) \arrow[ld, "h_{2*}"] \\
& \mathrm{Cay}(\mathbb{Z}_n, Y) &
\end{tikzcd}$$

FIG. 1. *Two isomorphic typical circulant ℓ-fold coverings for $\ell = \ell_1\ell_2$.*

a covering isomorphism $\Phi : \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_1) \to \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_2)$, then $f_{2*} = h_{2*} \circ \Phi'$ and $\Phi' \circ f_{1*} = h_{1*} \circ \Phi$. Thus,

$$h_* \circ \Phi = h_{2*} \circ h_{1*} \circ \Phi = h_{2*} \circ \Phi' \circ f_{1*} = f_{2*} \circ f_{1*} = f_*.$$

So, f_* and h_* are isomorphic through the covering isomorphism Φ.

Suppose that two typical circulant ℓ-fold coverings $f_* : \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_1) \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ and $h_* : \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_2) \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ are isomorphic through a covering isomorphism $\Phi : \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_1) \to \mathrm{Cay}(\mathbb{Z}_{\ell n}, X_2)$. Define $\Phi' : \mathbb{Z}_{\ell_2 n} \to \mathbb{Z}_{\ell_2 n}$ by $\Phi'(b) = \Phi(b) \pmod{\ell_2 n}$ for any $b \in \mathbb{Z}_{\ell_2 n}$. Then Φ' is an automorphism of the group $\mathbb{Z}_{\ell_2 n}$ since the restriction of Φ on $\mathbb{Z}_{\ell n}$ is an automorphism of the group $\mathbb{Z}_{\ell n}$. Thus, for any $a \in \mathbb{Z}_{\ell n}$, we have

$$\Phi' \circ f_{1*}(a) = \Phi'(a \pmod{\ell_2 n}) = \Phi(a) \pmod{\ell_2 n} = h_{1*} \circ \Phi(a),$$

which gives $\Phi' \circ f_{1*} = h_{1*} \circ \Phi$. Therefore $\Phi'(Z_1) = \Phi'(f_{1*}(X_1)) = h_{1*}(\Phi(X_1)) = h_{1*}(X_2) = Z_2$. So Φ' is also a graph isomorphism from $\mathrm{Cay}(\mathbb{Z}_{\ell_2 n}, Z_1)$ to $\mathrm{Cay}(\mathbb{Z}_{\ell_2 n}, Z_2)$. Moreover, for any $b \in \mathbb{Z}_{\ell_2 n}$, $h_{2*} \circ \Phi'(b) = h(1) \cdot \Phi'(b) = h_* \circ \Phi(b) = f_*(b) = f_{2*}(b)$, *i.e.*, $h_{2*} \circ \Phi' = f_{2*}$. Hence, f_{2*} and h_{2*} are isomorphic through the covering isomorphism Φ'. We know that $\Phi' \circ f_{1*} = h_{1*} \circ \Phi$ is also a typical circulant covering from $\mathrm{Cay}(\mathbb{Z}_{\ell n}, X_1)$ to $\mathrm{Cay}(\mathbb{Z}_{\ell_2 n}, Z_2)$, so $\Phi' \circ f_{1*}$ and h_{1*} are isomorphic through the covering isomorphism Φ (see Figure 1). □

By using Lemma 11, one can prove the following theorem.

THEOREM 12. *Let $\ell = \ell_1\ell_2$ be any composite number and let G be a connected circulant graph. If $\widetilde{G}$ is any typical circulant connected ℓ_2-fold covering of G, then the number of isomorphism classes of typical circulant connected ℓ-fold coverings of G is the product of the number of isomorphism classes of typical circulant connected ℓ_1-fold coverings of the graph $\widetilde{G}$ and the number of isomorphism classes of typical circulant connected ℓ_2-fold coverings of G.*

If ℓ is even and the valency of the circulant graph G is odd, then G has no typical circulant ℓ-fold covering because G has no typical circulant double covering. By using Theorems 1, 8, and 12 repeatedly, one can now enumerate the isomorphism classes of any finite-fold typical circulant connected coverings of a circulant graph.

Theorem 13. *Let $\ell = p_1^{r_1} p_2^{r_2} \cdots p_s^{r_s}$ be the prime factorization of any positive integer ℓ, and let G be a circulant graph of order n and valency d. Then the number N of isomorphism classes of typical circulant connected ℓ-fold coverings of G is as follows:*

$$N = \begin{cases} 0 & \text{if } \ell \text{ is even and } d \text{ is odd}, \\ \displaystyle\prod_{i=1}^{s} N_i & \text{otherwise}, \end{cases}$$

where

$$N_i = \begin{cases} p_i^{r_i(\lfloor \frac{d}{2} \rfloor - 1)} & \text{if } p_i | n, \\ p_i^{(r_i-1)(\lfloor \frac{d}{2} \rfloor - 1)}(p_i^{\lfloor \frac{d}{2} \rfloor} - 1) & \text{if } (p_i, n) = 1. \end{cases}$$

Note that the number N of isomorphism classes of typical circulant connected ℓ-fold coverings of G in Theorem 13 depends only on the folding number ℓ, the order n, and the valency d of the base graph $G = \mathrm{Cay}(\mathbb{Z}_n, Y)$. It means that it is independent of the choice of a set Y of connectors if $|Y| = d$.

Corollary 14. *Let G be a connected circulant graph of order n and valency d. For any prime p and any natural number r, the number of isomorphism classes of typical circulant connected p^r-fold coverings of G is 0 when d is odd and $p = 2$. Otherwise, this number is $p^{r(\lfloor \frac{d}{2} \rfloor - 1)}$ if $p|n$, and is $p^{(r-1)(\lfloor \frac{d}{2} \rfloor - 1)}(p^{\lfloor \frac{d}{2} \rfloor} - 1)$ if $(p, n) = 1$.*

Corollary 15. *Let G be a connected circulant graph of order n and valency d. For any two distinct primes p and q, the number N of isomorphism classes of typical circulant connected pq-fold coverings of G is 0 when d is odd and one of p and q is 2. Otherwise, the number N is*

$$N = \begin{cases} p^{\lfloor \frac{d}{2} \rfloor - 1} q^{\lfloor \frac{d}{2} \rfloor - 1} & \text{if } pq|n, \\ p^{\lfloor \frac{d}{2} \rfloor - 1}(q^{\lfloor \frac{d}{2} \rfloor} - 1) & \text{if } p|n \text{ but } (q, n) = 1, \\ (p^{\lfloor \frac{d}{2} \rfloor} - 1) q^{\lfloor \frac{d}{2} \rfloor - 1} & \text{if } q|n \text{ but } (p, n) = 1, \\ (p^{\lfloor \frac{d}{2} \rfloor} - 1)(q^{\lfloor \frac{d}{2} \rfloor} - 1) & \text{if } (pq, n) = 1. \end{cases}$$

For $G = K_n$, the complete graph of order n, the following result follows immediately.

Corollary 16. *Let $\ell = p_1^{r_1} p_2^{r_2} \cdots p_s^{r_s}$ be the prime factorization of any positive integer ℓ. Then there is no typical circulant connected ℓ-fold covering of K_n when both ℓ and n are even, otherwise the number of isomorphism classes of typical circulant connected ℓ-fold coverings of K_n is $\prod_{i=1}^{s} N_i$, where*

$$N_i = \begin{cases} p_i^{r_i(\lfloor \frac{n-1}{2} \rfloor - 1)} & \text{if } p_i | n, \\ p_i^{(r_i-1)(\lfloor \frac{n-1}{2} \rfloor - 1)}(p_i^{\lfloor \frac{n-1}{2} \rfloor} - 1) & \text{if } (p_i, n) = 1. \end{cases}$$

The number of isomorphism classes of typical circulant connected ℓ-fold coverings of the complete graph K_n for small ℓ and n is listed in Table 1.

5. Coverings of a trivalent circulant graph. Recall [5] that a circulant covering of a circulant graph is not necessarily a typical one. It is natural to ask which circulant coverings of a circulant graph are typical. A partial answer to this question will be given in this section.

TABLE 1

The number of isomorphism classes of typical circulant connected ℓ-fold coverings of the complete graph K_n for small ℓ and small n.

n	3	4	5	6	7	8	9	10	11	12	13	14	$\cdots$
$\ell=2$	1	0	3	0	7	0	15	0	31	0	63	0	
$\ell=3$	1	2	8	3	26	26	27	80	242	81	728	728	
$\ell=4$	1	0	6	0	28	0	120	0	496	0	2016	0	$\cdots$
$\ell=5$	4	4	5	24	124	124	624	125	3124	3124	15624	15624	
$\ell=6$	1	0	24	0	182	0	405	0	7502	0	45864	0	
$\vdots$							$\vdots$						

Let $G = \mathrm{Cay}(\mathbb{Z}_n, Y)$ be a trivalent circulant graph with $Y = \{\pm i, \frac{n}{2}\}$. One can check easily that if $G \neq K_4$ or $K_{3,3}$ then any edge of G determined by the connector i is contained in a unique 4-cycle but an edge determined by the connector $\frac{n}{2}$ is contained in exactly two 4-cycles.

Let $G = \mathrm{Cay}(\mathbb{Z}_n, Y)$ be neither K_4 nor $K_{3,3}$ and let $p : \mathrm{Cay}(\mathbb{Z}_{\ell n}, X) \to \mathrm{Cay}(\mathbb{Z}_n, Y)$ be a circulant connected ℓ-fold covering with $X = \{\pm j, \frac{\ell n}{2}\}$. Then p maps a 4-cycle in the covering graph $\mathrm{Cay}(\mathbb{Z}_{\ell n}, X)$ to a 4-cycle in G. Therefore, any edge $(a, a + \frac{\ell n}{2})$ of the covering graph determined by the connector $\frac{\ell n}{2}$ is mapped to an edge of G determined by the connector $\frac{n}{2}$. That is, for any $a \in \mathbb{Z}_{\ell n}$, we have $p(a + \frac{\ell n}{2}) = p(a) + \frac{n}{2}$. Since the labels of the vertices of the graph G can be translated, without any loss of generality, one can assume that $p(0) = 0$ and then $p(\frac{\ell n}{2}) = \frac{n}{2}$. Thus $p(N(0)) = N(0)$ which gives $p(\{j, -j\}) = \{i, -i\}$. If $p(j) = i$, then $p(N(j)) = N(i)$. Since $p(0) = 0$ and $p(j + \frac{\ell n}{2}) = p(j) + \frac{n}{2} = i + \frac{n}{2}$, we have $p(2j) = 2i$. By the same process, one can show that $p(\alpha j) = \alpha i$ for any integer α. For any $a_k = \alpha_k j + \varepsilon_k \frac{\ell n}{2} \in \mathbb{Z}_{\ell n}$, where $\varepsilon_k \in \mathbb{F}_2 = \{0, 1\}$, $k = 1, 2$,

$$p(a_1 + a_2) = p\left((\alpha_1 + \alpha_2)j + (\varepsilon_1 + \varepsilon_2)\frac{\ell n}{2}\right) = (\alpha_1 + \alpha_2)i + (\varepsilon_1 + \varepsilon_2)\frac{n}{2} = p(a_1) + p(a_2).$$

So p is an epimorphism from $\mathbb{Z}_{\ell n}$ onto $\mathbb{Z}_n$. Similarly, one can show that p is an epimorphism too if $p(j) = -i$. This proves the following theorem.

THEOREM 17. *Let G be a trivalent circulant graph but $G \neq K_4$ or $K_{3,3}$. Then every circulant connected covering of G is typical.*

The following corollary is immediate from Theorems 13 and 17.

COROLLARY 18. *Let $\ell = p_1^{r_1} p_2^{r_2} \cdots p_s^{r_s}$ be the prime factorization of a positive integer ℓ, and let G be a trivalent circulant graph of order n but $G \neq K_4$ or $K_{3,3}$. If ℓ is even, then G has no circulant connected ℓ-fold covering. If ℓ is odd, then the number of isomorphism classes of circulant connected ℓ-fold coverings of G is $\prod_{i=1}^{s} N_i$, where $N_i = 1$ or $p_i - 1$ corresponding to $p_i | n$ or $(p_i, n) = 1$, respectively.*

Example 1. Let $G_1 = K_4 = \mathrm{Cay}(\mathbb{Z}_4, \{1, 2, 3\})$ and $G_2 = K_{3,3} = \mathrm{Cay}(\mathbb{Z}_6, \{1, 3, 5\})$. Define covering projections $p_1 : \mathrm{Cay}(\mathbb{Z}_{12}, \{1, 6, 11\}) \to G_1$ by $p_1(4k) = 0$, $p_1(4k+1) = 2$, $p_1(4k+2) = 3$ and $p_1(4k+3) = 1$ for $k = 0, 1, 2$, and $p_2 : \mathrm{Cay}(\mathbb{Z}_{18}, \{1, 9, 17\}) \to G_2$ by $p_2(6k) = 0$, $p_2(6k+1) = 3$, $p_2(6k+2) = 4$, $p_2(6k+3) = 1$, $p_2(6k+4) = 2$, and $p_2(6k+5) = 5$ for $k = 0, 1, 2$. Then both p_1 and p_2 are circulant covering projections with connected covering graphs. Since $p_1(1+1) = 3$ and $p_1(1) + p_1(1) = 2 + 2 = 0$, p_1 is not a group homomorphism. Thus the covering projection p_1 is not typical. Similarly, one can show that the covering projection p_2 is not typical either. Furthermore, it can be checked that neither p_1 nor p_2 is isomorphic to a typical circulant covering projection.

REFERENCES

[1] J.-C. Bermond, F. Comellas, and D. F. Hsu, *Distributed loop computer networks: A survey*, J. Parallel Distrib. Comput., 24 (1995), pp. 2–10.

[2] S. F. Du, D. Marušič, and A. O. Waller, *On 2-arc-transitive covers of complete graphs*, J. Combin. Theory Ser. B, 74 (1998), pp. 276–290.

[3] S. F. Du, J. H. Kwak, and M. Y. Xu, *2-arc-transitive regular covers of complete graphs having the covering transformation group* $\mathbb{Z}_p^3$, J. Combin. Theory Ser. B, 93 (2005), pp. 73–93.

[4] R. Feng, J. H. Kwak, J. Kim, and J. Lee, *Isomorphism classes of concrete graph coverings*, SIAM J. Discrete Math., 11 (1998), pp. 265–272.

[5] R. Feng and J. H. Kwak, *Typical circulant double coverings of a circulant graph*, Discrete Math., 277 (2004), pp. 73–85.

[6] C. D. Godsil and A. D. Hensel, *Distance regular covers of the complete graph*, J. Combin. Theory Ser. B, 56 (1992), pp. 205–238.

[7] J. L. Gross and T. W. Tucker, *Generating all graph coverings by permutation voltage assignments*, Discrete Math., 18 (1977), pp. 273–283.

[8] M. Hofmeister, *Graph covering projections arising from finite vector spaces over finite fields*, Discrete Math., 143 (1995), pp. 87–97.

[9] M. Hofmeister, *Enumeration of concrete regular covering projections*, SIAM J. Discrete Math., 8 (1995), pp. 51–61.

[10] M. Hofmeister, *A note on counting connected graph covering projections*, SIAM J. Discrete Math., 11 (1998), pp. 286–292.

[11] J. H. Kwak, J. Chun, and J. Lee, *Enumeration of regular graph coverings having finite abelian covering transformation groups*, SIAM J. Discrete Math., 11 (1998), pp. 273–285.

[12] J. H. Kwak and J. Lee, *Isomorphism classes of graph bundles*, Canad. J. Math., 42 (1990), pp. 747–761.

[13] J. H. Kwak and J. Lee, *Enumeration of graph coverings, surface branched coverings and related group theory.* in Combinatorial and Computational Mathematics, S. Hong, J. H. Kwak, K. H. Kim, and F. W. Raush, eds., World Scientific, Rivers Edge, NJ, 2001, pp. 97–161.

[14] J. H. Park and K. Y. Chwa, *Recursive circulant: A new topology for multicomputer networks*, in Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN'94), IEEE Press, New York, 1994, pp. 73–80.

[15] J. Siagiova, *Composition of regular coverings of graphs*, J. Electrical Engrg, 50 (1999), pp. 75–77.

[16] J. Siagiova, *Composition of regular coverings of graphs and voltage assignments*, Australas. J. Combin., 28 (2003), pp. 131–136.

SIAM J. Discrete Math.
Vol. 21, No. 2, pp. 548–550

ERRATUM: ENUMERATING TYPICAL CIRCULANT COVERING PROJECTIONS ONTO A CIRCULANT GRAPH*

RONGQUAN FENG[†], JIN HO KWAK[‡], AND YOUNG SOO KWON[§]

Abstract. This paper consists of an erratum to the previously published *Enumerating Typical Circulant Covering Projections onto a Circulant Graph.*

DOI. 10.1137/060653020

In Lemma 4 of [1], the authors incorrectly rephrased the characterization theorem of two isomorphic graph coverings given in [3] for a regular covering case. This should be corrected as follows.

Lemma 4′. *Let ϕ and ψ be typical voltage assignments in $C^1(G;\mathbb{Z}_p)$. Then, two typical circulant p-fold coverings $p_\phi : G^\phi \to G$ and $p_\psi : G^\psi \to G$ are isomorphic if and only if there exists a function $g : V(G) \to S_p$ such that $\psi(uv) = g(v)\phi(uv)g(u)^{-1}$ in S_p for each $uv \in D(G)$, where S_p is the symmetric group on the elements of $\mathbb{Z}_p$ and $\mathbb{Z}_p$ is considered as the left regular subgroup of S_p.*

However, if the voltage assignments ϕ and ψ in $C^1(G;\mathbb{Z}_p)$ are assumed to be trivial on a spanning tree of a graph G, two coverings $p_\phi : G^\phi \to G$ and $p_\psi : G^\psi \to G$ are isomorphic if and only if there exists an automorphism $\sigma \in \operatorname{Aut}(\mathbb{Z}_p)$ such that $\phi(uv)^\sigma = \psi(uv)$ for every arc uv of G (see [2, 4]).

Because of the error in Lemma 4, Lemma 5 of [1] is also incorrect. Instead of these two lemmas, we use a minor extension of Lemma 10 for our enumeration, which can be stated as follows.

Lemma 10′. *Let $G = \operatorname{Cay}(\mathbb{Z}_n, Y)$ be a connected circulant graph. For any natural number ℓ (not necessarily prime), let $f, g : \mathbb{Z}_{\ell n} \to \mathbb{Z}_n$ be two group epimorphisms. Then two connected typical coverings $f_* : \operatorname{Cay}(\mathbb{Z}_{\ell n}, X_1) \to G$ and $g_* : \operatorname{Cay}(\mathbb{Z}_{\ell n}, X_2) \to G$ are isomorphic if and only if there exists an automorphism $\Phi \in \operatorname{Aut}(\mathbb{Z}_{\ell n})$ such that $g \circ \Phi = f$ and $\Phi(X_1) = X_2$.*

The necessity of Lemma 10′ is proved in [1] and the sufficiency is clear.

Based on Lemmas 2, 3, and 7 in [1] and Lemma 10′, Theorem 8 in [1], which counts the connected typical circulant prime-fold coverings, should be corrected as follows.

Theorem 8′. *For any odd prime p, the number of isomorphism classes of connected typical circulant p-fold coverings of $G = \operatorname{Cay}(\mathbb{Z}_n, Y)$ is $\frac{1}{p-1}(p^{\lfloor \frac{d}{2} \rfloor} - 1)$ if $(p, n) = 1$ and is $p^{\lfloor \frac{d}{2} \rfloor - 1}$ otherwise, where d is the valency of G.*

Proof. Let $Y = \{\pm i_1, \pm i_2, \ldots, \pm i_k\}$ or $\{\pm i_1, \pm i_2, \ldots, \pm i_k, \frac{n}{2}\}$ according to whether the valency d of G is even $2k$ or odd $2k+1$, where $0 < i_1, i_2, \ldots, i_k < \lfloor \frac{n+1}{2} \rfloor$. Then, by Lemma 3, any typical circulant p-fold covering of G can be derived from a typical voltage assignment, or from a k-tuple $(\delta_1, \delta_2, \ldots, \delta_k) \in \mathbb{Z}_p^k$, with the assumption that any typical covering projection sends 1 in $\mathbb{Z}_{pn}$ to 1 in $\mathbb{Z}_n$ by Lemma 2. Let Δ

*Received by the editors February 26, 2006; accepted for publication (in revised form) September 22, 2006; published electronically June 28, 2007.

http://www.siam.org/journals/sidma/21-2/65302.html

[†]LMAM, School of Mathematical Sciences, Peking University, Beijing 100871, People's Republic of China (fengrq@math.pku.edu.cn).

[‡]Department of Mathematics, Pohang University of Science and Technology, Pohang, 790-784 Korea (jinkwak@postech.ac.kr).

[§]Department of Mathematics, Yeungnam University, Kyeongsan, 712-749 Korea (ysookwon@ynu.ac.kr).

denote the set of k-tuples $(\delta_1, \delta_2, \ldots, \delta_k) \in \mathbb{Z}_p^k$ which induce connected p-fold coverings of G. Then $|\Delta| = p^{\lfloor \frac{d}{2} \rfloor} - 1$ if $(p, n) = 1$ and $|\Delta| = p^{\lfloor \frac{d}{2} \rfloor}$ if $p|n$ by Lemma 7. Furthermore, by Lemma 10$'$, any two k-tuples $(\delta_1, \delta_2, \ldots, \delta_k)$ and $(\delta_1', \delta_2', \ldots, \delta_k')$ in Δ induce isomorphic coverings if and only if there exists a $\Phi \in \mathrm{Aut}\,(\mathbb{Z}_{pn})$ such that $\Phi(i_j + \delta_j n) = i_j + \delta_j' n$ for every $j = 1, 2, \ldots, k$. In this case, Φ becomes a covering isomorphism between induced coverings and $\Phi(1) = 1 + an$ for some $a = 0, 1, \ldots, p-1$. Note that a map Φ defined by $\Phi(1) = 1 + an$ for some $a = 0, 1, \ldots, p-1$ is an automorphism of $\mathbb{Z}_{pn}$ if and only if $(pn, 1 + an) = 1$. Let $\mathcal{S} = \{\Phi \in \mathrm{Aut}\,(\mathbb{Z}_{pn}) \mid \Phi(1) \equiv 1 (\mathrm{mod}\ n)\}$. Then, $\mathcal{S}$ is a subgroup of $\mathrm{Aut}\,(\mathbb{Z}_{pn})$. Define an $\mathcal{S}$-action on Δ by $\Phi(\delta_1, \delta_2, \ldots, \delta_k) = (\delta_1', \delta_2', \ldots, \delta_k')$ for any $\Phi \in \mathcal{S}$ and $(\delta_1, \delta_2, \ldots, \delta_k) \in \Delta$, where δ_j' is uniquely determined by the relation $\Phi(i_j + \delta_j n) = i_j + \delta_j' n$ for every $j = 1, 2, \ldots, k$. This action is well defined and the number of isomorphism classes of connected typical circulant p-fold coverings of G is the number of orbits under the $\mathcal{S}$-action on Δ. Let an arbitrary $\delta = (\delta_1, \delta_2, \ldots, \delta_k) \in \Delta$ be given. Since no $\Phi \in \mathcal{S}$ fixes the k-tuple $(i_1 + \delta_1 n, i_2 + \delta_2 n, \ldots, i_k + \delta_k n)$ except the identity Φ, the orbit size of δ equals the cardinality $|\mathcal{S}|$, that is, the number of automorphisms Φ of $\mathbb{Z}_{pn}$ such that $\Phi(1) = 1 + an$ for some $a = 0, 1, \ldots, p-1$. As the first case, let $(p, n) = 1$. Then, $\gcd(pn, 1 + an) = 1$ except exactly one of $a = 0, 1, \ldots p-1$. Hence, the orbit size of δ is $p-1$. Since $\delta \in \Delta$ is given arbitrarily, it gives the proof of the case $(p, n) = 1$. As the remaining case, let $p|n$. Then, for each $a = 0, 1, \ldots p-1$, we get $\gcd(1 + an, pn) = 1$. Hence, the orbit size of δ is p. This completes the proof. □

Comparing with the old enumeration in Theorem 8 in [1], the number of isomorphism classes of connected typical circulant p-fold coverings of $G = \mathrm{Cay}(\mathbb{Z}_n, Y)$ is corrected as the multiple of the old value by $\frac{1}{p-1}$ when $(p, n) = 1$ in Theorem 8$'$. The following corrections will be listed as the last part of this manuscript.

Since Lemma 7 in [1] counts the number of disconnected typical circulant p-fold coverings of $G = \mathrm{Cay}(\mathbb{Z}_n, Y)$, one can get the number of isomorphism classes of (connected or not) typical circulant p-fold coverings of G with the help of Theorem 8$'$. This provides a correct version of Theorem 6 in [1].

Now, Theorem 13 in [1], which counts the connected typical circulant ℓ-fold coverings for any composite number ℓ, can be revised as follows.

Theorem 13$'$. *Let $\ell = p_1^{r_1} p_2^{r_2} \cdots p_s^{r_s}$ be the prime factorization of a positive integer ℓ and let G be a connected circulant graph of order n and valency d. Then the number N of isomorphism classes of connected typical circulant ℓ-fold coverings of G is*

$$N = \begin{cases} 0 & \text{if } \ell \text{ is even and } d \text{ is odd,} \\ \displaystyle\prod_{i=1}^{s} N_i & \text{otherwise,} \end{cases}$$

where

$$N_i = \begin{cases} p_i^{r_i(\lfloor \frac{d}{2} \rfloor - 1)} & \text{if } p_i | n, \\ p_i^{(r_i - 1)(\lfloor \frac{d}{2} \rfloor - 1)} \left(p_i^{\lfloor \frac{d}{2} \rfloor} - 1 \right) / (p_i - 1) & \text{if } (p_i, n) = 1. \end{cases}$$

Corollaries 14, 15, 16, and 18 and Table 1 should be revised as follows.

Corollary 14$'$. *Let G be a connected circulant graph of order n and valency d. For any prime p and any natural number r, the number of isomorphism classes*

Table 0.1

The number of isomorphism classes of connected typical circulant ℓ-fold coverings of the complete graph K_n for small ℓ and small n

n	3	4	5	6	7	8	9	10	11	12	13	14	$\cdots$
$\ell = 2$	1	0	3	0	7	0	15	0	31	0	63	0	$\cdots$
$\ell = 3$	1	1	4	3	13	13	27	40	121	81	364	364	
$\ell = 4$	1	0	6	0	28	0	120	0	496	0	2016	0	
$\ell = 5$	1	1	5	6	31	31	131	125	321	321	3906	3906	
$\ell = 6$	1	0	12	0	91	0	405	0	3751	0	22932	0	$\cdots$
$\vdots$	$\vdots$											$\vdots$	

of connected typical circulant p^r-fold coverings of G is 0 when $p = 2$ and d is odd. Otherwise, this number is $p^{r(\lfloor \frac{d}{2} \rfloor - 1)}$ if $p|n$, and is $p^{(r-1)(\lfloor \frac{d}{2} \rfloor - 1)}(p^{\lfloor \frac{d}{2} \rfloor} - 1)/(p-1)$ if $(p, n) = 1$.

Corollary 15′. *Let G be a connected circulant graph of order n and valency d. For any two distinct primes p and q, the number N of isomorphism classes of connected typical circulant pq-fold coverings of G is 0 when d is odd and one of p and q is 2. Otherwise, the number N is*

$$N = \begin{cases} p^{\lfloor \frac{d}{2} \rfloor - 1} q^{\lfloor \frac{d}{2} \rfloor - 1} & \text{if } pq|n, \\ p^{\lfloor \frac{d}{2} \rfloor - 1}(q^{\lfloor \frac{d}{2} \rfloor} - 1)/(q-1) & \text{if } p|n \text{ but } (q,n) = 1, \\ (p^{\lfloor \frac{d}{2} \rfloor} - 1) q^{\lfloor \frac{d}{2} \rfloor - 1}/(p-1) & \text{if } q|n \text{ but } (p,n) = 1, \\ (p^{\lfloor \frac{d}{2} \rfloor} - 1)(q^{\lfloor \frac{d}{2} \rfloor} - 1)/((p-1)(q-1)) & \text{if } (pq, n) = 1. \end{cases}$$

Corollary 16′. *Let $\ell = p_1^{r_1} p_2^{r_2} \cdots p_s^{r_s}$ be the prime factorization of a positive integer ℓ. Then no connected typical circulant ℓ-fold covering of K_n exists when both ℓ and n are even. Otherwise the number of isomorphism classes of connected typical circulant ℓ-fold coverings of K_n is $\prod_{i=1}^{s} N_i$, where*

$$N_i = \begin{cases} p_i^{r_i(\lfloor \frac{n-1}{2} \rfloor - 1)} & \text{if } p_i|n, \\ p_i^{(r_i - 1)(\lfloor \frac{n-1}{2} \rfloor - 1)} \left(p_i^{\lfloor \frac{n-1}{2} \rfloor} - 1 \right) / (p_i - 1) & \text{if } (p_i, n) = 1. \end{cases}$$

Corollary 18′. *Let G be a connected circulant trivalent graph of order n but $G \neq K_4$ or $K_{3,3}$. If ℓ is even, then G has no connected circulant ℓ-fold coverings. If ℓ is odd, then G has only one connected circulant ℓ-fold covering up to isomorphism.*

All statements in [1] that were not mentioned remain valid.

REFERENCES

[1] R. Feng, J. H. Kwak, and Y. S. Kwon, *Enumerating typical circulant covering projections onto a circulant graph*, SIAM J. Discrete Math., 19 (2005), pp. 196–207.

[2] S. Hong, J. H. Kwak, and J. Lee, *Regular graph coverings whose covering transformation groups have the isomorphism extension property*, Discrete Math., 148 (1996), pp. 85–105.

[3] J. H. Kwak and J. Lee, *Isomorphism classes of graph bundles*, Canad. J. Math., 42 (1990), pp. 747–761.

[4] M. Škoviera, *A contribution to the theory of voltage graphs*, Discrete Math., 61 (1986), pp. 281–292.

SIAM J. Discrete Math.
Vol. 19, No. 1, pp. 208–223

ON THE STRUCTURE OF GRAPHS WITH NON-SURJECTIVE $L(2,1)$-LABELINGS*

JOHN P. GEORGES† AND DAVID W. MAURO†

Abstract. For a graph G, an $L(2,1)$-labeling of G with span k is a mapping $L \to \{0,1,2,\dots,k\}$ such that adjacent vertices are assigned integers which differ by at least 2, vertices at distance two are assigned integers which differ by at least 1, and the image of L includes 0 and k. The minimum span over all $L(2,1)$-labelings of G is denoted $\lambda(G)$, and each $L(2,1)$-labeling with span $\lambda(G)$ is called a λ-labeling. For $h \in \{1,\dots,k-1\}$, h is a hole of L if and only if h is not in the image of L. The minimum number of holes over all λ-labelings is denoted $\rho(G)$, and the minimum k for which there exists a surjective $L(2,1)$-labeling onto $\{0,1,\dots,k\}$ is denoted $\mu(G)$. This paper extends the work of Fishburn and Roberts on ρ and μ through the investigation of an equivalence relation on the set of λ-labelings with ρ holes. In particular, we establish that $\rho \le \Delta$. We analyze the structure of those graphs for which $\rho \in \{\Delta-1,\Delta\}$, and we show that $\mu = \lambda+1$ whenever λ is less than the order of the graph. Finally, we give constructions of connected graphs with $\rho = \Delta$ and order $t(\Delta+1)$, $1 \le t \le \Delta$.

Key words. $L(2,1)$-labeling, λ-labeling, hole index, dominating vertex set

AMS subject classifications. 05C

DOI. 10.1137/S0895480103429800

1. Introduction.

The $L(2,1)$-labeling problem is a vertex-labeling analog of Hale's channel assignment problem [14] which seeks to minimize the range of frequencies used while at the same time ensuring that transmitters which are sufficiently close together are assigned transmission frequencies which differ by no less than a prescribed amount.

Let G be a simple graph with vertex set $V(G)$ and edge set $E(G)$. For fixed positive integer k, an $L(2,1)$*-labeling of* G with *span* k is a mapping L from $V(G)$ into $\{0,1,2,\dots,k\}$ such that any two vertices which are adjacent are assigned integers which differ by at least 2, any two vertices which are distance two apart are assigned integers which differ by at least 1, and the integers 0 and k are each assigned to at least one vertex. We denote the span k of L by $s(L)$, and for each vertex $v \in V(G)$, we refer to $L(v)$ as the label of v assigned by L. The minimum span among all $L(2,1)$-labelings of G is called the λ*-number of* G, denoted $\lambda(G)$. Any $L(2,1)$-labeling which achieves a span of $\lambda(G)$ is called a λ*-labeling of* G.

For an $L(2,1)$-labeling L of G and for integer h such that $0 < h < s(L)$, h *is a hole of* L if and only if h is not assigned by L to any vertex v in $V(G)$. The minimum number of holes over all λ-labelings of G is called the *hole index of* G, and is denoted $\rho(G)$. If there exists a λ-labeling L of G with no holes, then L is called a *no-hole* λ*-labeling of* G and G is said to be λ*-full-colorable*. Alternatively, G is λ-full-colorable if and only if there exists a surjective λ-labeling of G. If there exists an $L(2,1)$-labeling of G (not necessarily a λ-labeling) with no holes, then the minimum span over all such $L(2,1)$-labelings of G is denoted $\mu(G)$. Clearly, $\mu(G) \ge \lambda(G)$, and $\mu(G) = \lambda(G)$ if and only if $\rho(G) = 0$.

*Received by the editors June 16, 2003; accepted for publication (in revised form) October 26, 2004; published electronically July 18, 2005.
http://www.siam.org/journals/sidma/19-1/42980.html

†Department of Mathematics, Trinity College, Hartford, CT 06106 (john.georges@trincoll.edu, david.mauro@trincoll.edu).

The $L(2,1)$-labeling was introduced by Griggs and Yeh [13] as an extension of T-colorings (see [16]). There, they considered the λ-numbers of graphs in various classes such as trees, cycles, and paths, and they investigated the relationship between $\lambda(G)$ and other graph invariants such as $\Delta(G)$ and $\chi(G)$. Since then, many other authors have extended these lines of study, exploring the λ-numbers of the n-cube [19], chordal graphs [17], various products of graphs [10, 11, 15], as well as exploring the relationship between $\lambda(G)$ and other invariants such as the size of G [9] and the path covering number of G^c (the complement of G) [12]. Generalizations of $L(2,1)$-labelings have also been considered; see [2, 4, 8, 10, 11, 18].

Recently, attention has turned to the study of graphs G for which $\rho(G) = 0$. Fishburn and Roberts [6, 7] in particular have shown that $\rho(G) = 0$ if $|V(G)| = \lambda(G) + 1$, and that $\rho(G) = 0$ if G is any tree distinct from the claw $K_{1,n}$. They have constructed a number of graphs G with $\rho(G) > 0$, and, in the event that $\rho(G) > 0$, they have shown that $\lambda(G) + \rho(G)$ is an upper bound for $\mu(G)$ if $\mu(G)$ exists.

In this paper, we continue the study of $\rho(G)$ with emphasis on $\rho(G) > 0$. Section 2 provides notation, definitions, and an equivalence class on the set of λ-labelings of G with $\rho(G)$ holes which will facilitate our discussion. We consider the relationship between $\rho(G)$ and $\Delta(G)$ (section 3) and the relationships among $\rho(G), \mu(G)$, and $\lambda(G)$ (section 4). In section 5, we explore the structure of graphs with the property $\rho(G) = \Delta(G)$.

2. Definitions and preliminary results. The sum $G_1 + G_2$ of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the graph $G = (V, E)$ with $V = V_1 \bigcup V_2$ and $E = E_1 \bigcup E_2$.

Let L be an $L(2,1)$-labeling of G. Then $M_i(G, L) = \{v \in V(G) | L(v) = i\}$ and $m_i(G, L) = |M_i(G, L)|$.

Let L be a λ-labeling of G. Suppose $0 < h_1 < h_2 < h_3 < \cdots < h_w < \lambda(G)$ are the holes of L. Then for k, $1 \le k \le w - 1$, the set of integers strictly between h_k and h_{k+1} shall be called *island* k *of* L, denoted $I_k(L)$. Similarly, *island* 0 *of* L, denoted $I_0(L)$, and *island* w *of* L, denoted $I_w(L)$, shall, respectively, mean $\{0, 1, 2, \ldots, h_1 - 1\}$ and $\{h_w + 1, h_w + 2, \ldots, \lambda(G)\}$. For $0 \le k \le w$, the smallest element of $I_k(L)$ shall be called the *left coast of* $I_k(L)$ (denoted $\mathrm{lc}(I_k(L))$) and the largest element of $I_k(L)$ shall be called the *right coast of* $I_k(L)$ (denoted $\mathrm{rc}(I_k(L))$). Integers which are the left coast or right coast of some island will be called *coastal labels.* The *interior of* $I_k(L)$, denoted $\mathrm{int}(I_k(L))$, shall mean $I_k(L) - \{\mathrm{lc}(I_k(L)), \mathrm{rc}(I_k(L))\}$. The set of coastal labels in island $I_k(L)$ will be denoted $C(I_k(L))$. In the case of the equivalent conditions $|C(I_k(L))| = 1, |I_k(L)| = 1$, and $\mathrm{lc}(I_k(L)) = \mathrm{rc}(I_k(L))$, we shall refer to $I_k(L)$ as an *atoll.*

For any island $I_j(L) = \{x, x+1, \ldots, x+z\}$, we let $Z_j(L)$ denote the sequence of sets of vertices $(M_x(G, L), M_{x+1}(G, L), \ldots, M_{x+z}(G, L))$. We also define $Z(L)$ to be the sequence $(Z_0(L), Z_1(L), Z_2(L), \ldots, Z_w(L))$.

For any graph G, let $\Lambda_\rho(G)$ be the collection of all λ-labelings of G with $\rho(G)$ holes. Also, let $\mathcal{L}(G, t)$ be the collection of $L(2,1)$-labelings of G with span t. It is clear that if $L \in \mathcal{L}(G, t)$, then the labeling $L' = t - L$ is also in $\mathcal{L}(G, t)$. We therefore observe that $v \in M_i(G, L)$ if and only if $v \in M_{t-i}(G, L')$.

We next define and illustrate two classes of vertex labelings of G, elements of which follow from a given labeling $L \in \Lambda_\rho(G)$.

For any $L \in \Lambda_\rho(G)$ and any island $I_j(L)$, define

$$\phi_j(L)(v) = \begin{cases} L(v) & \text{if } L(v) \notin I_j(L), \\ \mathrm{rc}(I_j(L)) - i & \text{if } L(v) = \ \mathrm{lc}(I_j(L)) + i \in I_j(L). \end{cases}$$

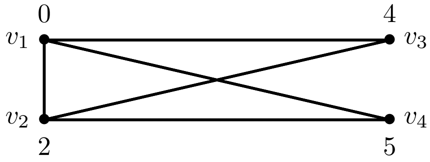


FIG. 2.1. *$L(2,1)$-labeling of $K_{1,1,2}$.*

We call this labeling of the vertices of G an *intra-island relabeling at L,* and note that $\phi_j(L)$ is easily seen to be an element of $\Lambda_\rho(G)$ with holes identical to the holes of L. It therefore follows that the composition of any number of intra-island relabelings at L is an element of $\Lambda_\rho(G)$. We observe that the components of $Z_j(\phi_j(L))$ are the components of $Z_j(L)$ in opposite order. (For $k \neq j$, $Z_k(\phi_j(L)) = Z_k(L)$.) We also observe that the relation Φ on $\Lambda_\rho(G)$, given by $(L_1, L_2) \in \Phi$ if and only if L_2 is a finite composition of intra-island relabelings at L_1, is an equivalence relation. Moreover, the cardinality of the equivalence class containing L is $2^{\rho(G)+1-a}$, where a is the number of atolls of L.

For any $L \in \Lambda_\rho(G)$ and for a fixed j, $0 \leq j \leq \rho(G) - 1$, define

$$\psi_j(L)(v) = \begin{cases} L(v) & \text{if } L(v) \notin I_j(L) \bigcup I_{j+1}(L) \\ L(v) - (\mathrm{lc}(I_{j+1}(L)) - \mathrm{lc}(I_j(L))) & \text{if } L(v) \in I_{j+1}(L) \\ L(v) + \mathrm{rc}(I_{j+1}(L)) - \mathrm{lc}(I_{j+1}(L)) + 2 & \text{if } L(v) \in I_j(L). \end{cases}$$

We call this labeling of G an *inter-island relabeling at L,* and note that $\psi_j(L)$ is an element of $\Lambda_\rho(G)$ with the following properties:

1. $\psi_j(L)$ has a hole at $\mathrm{lc}(I_j(L)) + \mathrm{rc}(I_{j+1}(L)) - \mathrm{lc}(I_{j+1}(L)) + 1$;
2. $Z_{j+1}(\psi_j(L)) = Z_j(L)$;
3. $Z_j(\psi_j(L)) = Z_{j+1}(L)$.

We also note that since $\psi_j(L) \in \Lambda_\rho(G)$, it follows that the composition of any finite number of inter-island relabelings at L is an element of $\Lambda_\rho(G)$ as well.

Example 2.1. Consider the graph $G = K_{1,1,2}$ along with an $L(2,1)$-labeling L as given in Figure 2.1.

Since it is easily seen that $\lambda(G) = 5$ and $\rho(G) = 2$, then $L \in \Lambda_\rho(G)$ with islands $I_0(L) = \{0\}$, $I_1(L) = \{2\}$ and $I_2(L) = \{4, 5\}$. Thus,

$$\psi_1(L)(v) = \begin{cases} 0 & \text{if } v = v_1, \\ 5 & \text{if } v = v_2, \\ 2 & \text{if } v = v_3, \\ 3 & \text{if } v = v_4, \end{cases}$$

and the islands of $\psi_1(L)$ are $\{0\}, \{2, 3\}$, and $\{5\}$.

Additionally,

$$\phi_2(L)(v) = \begin{cases} 0 & \text{if } v = v_1, \\ 2 & \text{if } v = v_2, \\ 5 & \text{if } v = v_3, \\ 4 & \text{if } v = v_4. \end{cases}$$

We next note that for any finite composition $\psi(L)$ of inter-island relabelings at L, there exists a permutation θ of $\{0, 1, 2, \ldots, \rho(G)\}$ such that

$$Z(\psi(L)) = (Z_{\theta^{-1}(0)}(L), Z_{\theta^{-1}(1)}(L), \ldots, Z_{\theta^{-1}(\rho(G))}(L)).$$

And, conversely, for every permutation θ of $\{0, 1, 2, \ldots, \rho(G)\}$, there exists a finite composition $\psi(L)$ of inter-island relabelings at L such that $Z(\psi(L)) = (Z_{\theta^{-1}(0)}(L), Z_{\theta^{-1}(1)}(L), \ldots, Z_{\theta^{-1}(\rho(G))}(L))$. It follows that for any $L \in \Lambda_\rho(G)$ with islands $I_0(L), I_1(L), \ldots, I_{\rho(G)}(L)$, there is a composition ψ of inter-island relabelings at L with islands $I_0(\psi(L)), I_1(\psi(L)), \ldots, I_{\rho(G)}(\psi(L))$ such that $|I_0(\psi(L))| \le |I_1(\psi(L))| \le \cdots \le |I_{\rho(G)}(\psi(L))|$. Thus, without losing the generality of G, we shall assume $|I_0(L)| \le |I_1(L)| \le \cdots \le |I_{\rho(G)}(L)|$ when convenient.

Example 2.2. Let G be a graph with $\rho(G) = 2$ and let $L \in \Lambda_\rho(G)$. Let $\psi(L) = \psi_0 \circ \psi_1(L)$. Then

$$Z(L) = (Z_0(L), Z_1(L), Z_2(L))$$

and

$$Z(\psi(L)) = (Z_2(L), Z_0(L), Z_1(L)).$$

It is easy to see that the relation Ψ on $\Lambda_\rho(G)$, given by $(L_1, L_2) \in \Psi$ if and only if $L_2 = \psi(L_1)$ for some finite composition ψ of inter-island relabelings at L_1, is an equivalence relation. Moreover, the cardinality of each equivalence class under Ψ is $(\rho(G)+1)!$.

Finally, we observe that the relation Ω on $\Lambda_\rho(G)$, given by $(L_1, L_2) \in \Omega$ if and only if $L_2 = \omega(L_1)$ for some finite composition ω of inter- and/or intra-island relabelings at L_1, is an equivalence relation, and that there are $(\rho(G)+1)!2^{\rho(G)+1-a}$ members in each equivalence class containing L_1, where a is the number of atolls of L_1.

Example 2.3. If $G = K_{2,3}$, then $\lambda(G) = 5$ and $\rho(G) = 1$. Furthermore, every λ-labeling of G is in $\Lambda_\rho(G)$, each such labeling induces 2 islands (one with cardinality two and one with cardinality three), and $|\Lambda_\rho(G)| = 24$. Finally, for $L \in \Lambda_\rho(G)$, $|[L]_\Phi| = 4$, $|[L]_\Psi| = 2$, and $|[L]_\Omega| = 8$.

Example 2.4. If $G = K_2 + K_4$, then $\lambda(G) = 6$ and $\rho(G) = 1$. The graph G has 720 different λ-labelings, of which 144 are in $\Lambda_\rho(G)$. Among the islands in $\Lambda_\rho(G)$, 48 induce 2 islands of cardinality 3 each, and the other 96 labelings induce 2 islands with cardinalities 1 and 5. We are not aware of the existence of a connected graph having $\rho(G) \ge 1$ which has two labelings which induce islands having different cardinalities as illustrated in the analysis of the disconnected graph $K_2 + K_4$.

We close this section with a definition and related theorem which will prove useful in section 4.

Let H be a graph. Then a *path covering of* H is a set of vertex-disjoint paths in H which cover $V(H)$. The path-covering number of H, denoted $c(H)$, is the minimum cardinality over all path coverings of H.

THEOREM 2.5 ([12]). *Suppose G is a graph with $|V(G)| = n$. Then*

1. $\lambda(G) = n + c(G^c) - 2$ *if* $c(G^c) \ge 2$
2. $\lambda(G) \le n - 1$ *if* $c(G^c) = 1$.

3. Relating $\rho(G)$ and $\Delta(G)$.

In this section, we make use of configurations of islands to explore the relationship between $\rho(G)$ and $\Delta(G)$.

LEMMA 3.1. *Let G be a graph with $\rho(G) \ge 1$, let $L \in \Lambda_\rho(G)$ and let $0 \le i < j \le \rho(G)$. Suppose $x \in \{lc(I_i(L)), rc(I_i(L))\}$ and $y \in \{lc(I_j(L)), rc(I_j(L))\}$. Then*

1. *for each $v \in M_x(G, L)$, there exists a unique vertex $w \in M_y(G, L)$ such that w and v are adjacent, and*
2. $m_x(G, L) = m_y(G, L)$.

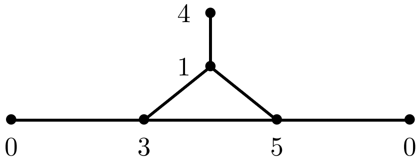


FIG. 3.1. *Graph G with $\rho(G) = 0$.*

Proof. Through some finite composition ω of inter- and/or intra-island relabelings at L, we may construct an element $\omega(L)$ of $\Lambda_\rho(G)$ such that for some α, α is a hole of $\omega(L)$, $M_x(G,L) = M_{\alpha-1}(G,\omega(L))$, and $M_y(G,L) = M_{\alpha+1}(G,\omega(L))$.

Proof of (1). Select $v \in M_{\alpha-1}(G,\omega(L))$, and suppose to the contrary that for every vertex $w \in M_{\alpha+1}(G,\omega(L))$, $\{v,w\} \notin E(G)$. Select vertex $w' \in M_{\alpha+1}(G,\omega(L))$. If $|M_{\alpha+1}(G,\omega(L))| \geq 2$, we produce an $L(2,1)$-labeling L' of G with $\rho(G)-1$ holes

$$L'(u) = \begin{cases} \omega(L)(u) & \text{if } u \neq w', \\ \omega(L)(u) - 1 & \text{if } u = w', \end{cases}$$

contradicting that $\omega(L)$ is a λ-labeling with the minimum number of holes. On the other hand, if $|M_{\alpha+1}(G,\omega(L))| = 1$, then we produce an $L(2,1)$-labeling L' of G with span $\lambda(G)-1$,

$$L'(u) = \begin{cases} \omega(L)(u) & \text{if } \omega(L)(u) \leq \alpha - 1, \\ \omega(L)(u) - 1 & \text{otherwise,} \end{cases}$$

contradicting that $\omega(L)$ is a λ-labeling. Thus, for each $v \in M_x(G,L)$, there exists vertex $w \in M_y(G,L)$ such that w and v are adjacent. Uniqueness of w follows from the distance 2 condition.

Proof of (2) follows immediately from (1). □

Example 3.2. Consider the graph G and $L(2,1)$-labeling L of G given in Figure 3.1. It is easily verified that L is a λ-labeling of G with one hole at 2; hence $\rho(G) \leq 1$. Since $1 = m_5(G,L) \neq m_0(G,L) = 2$, Lemma 3.1 implies that $\rho(G) < 1$. Hence, there must exist a λ-labeling of G with $\rho(G) = 0$.

When there is no chance of confusion, we may hereafter suppress the functional dependence of the various island notations on L. Likewise, we may suppress the functional dependence of the notations $M_i(G,L)$ and $m_i(G,L)$ on G and L.

LEMMA 3.3. *Let G be a graph with $\rho(G) \geq 1$ and let $L \in \Lambda_\rho(G)$. Then $\Delta(G) \geq \sum_{j=1}^{\rho(G)} |C(I_j)| \geq \rho(G)$.*

Proof. Let v be a vertex with label $\mathrm{rc}(I_0)$ under L. Then from Lemma 3.1, it follows that for $1 \leq j \leq \rho(G)$ and for y in $\{\mathrm{lc}(I_j),\mathrm{rc}(I_j)\}$, v is adjacent to some vertex in M_y. Thus, $\Delta(G) \geq d(v) \geq \sum_{j=1}^{\rho(G)} |C(I_j)| \geq \sum_{j=1}^{\rho(G)} 1 = \rho(G)$. □

Recalling that L exists in $\Lambda_\rho(G)$ such that $|I_0| \leq |I_1| \leq |I_2| \leq \cdots \leq |I_{\rho(G)}|$, we note that the greatest lower bound for $\Delta(G)$ afforded by $\sum_{j=1}^{\rho(G)} |C(I_j)|$ occurs at such L. We also note that the following result is an immediate consequence of Lemma 3.3.

THEOREM 3.4. *For any graph G, $\rho(G) \leq \Delta(G)$.*

For the remainder of this section, we shall consider the structures of graphs associated with $\rho(G) = \Delta(G)$ and $\rho(G) = \Delta(G) - 1$, with particular attention paid to Δ-regular graphs.

THEOREM 3.5. *Let G be a graph with $\rho(G) = \Delta(G)$ and let $L \in \Lambda_\rho(G)$. Then*

1. *every island of L is an atoll; particularly, $I_j = \{2j\}$ for $0 \leq j \leq \Delta(G)$.*

2. *$\lambda(G) = 2\Delta(G)$.*
3. *G is Δ-regular and $|V(G)| \equiv 0$ mod $(\Delta(G)+1)$.*
4. *For every j, $0 \leq j \leq \Delta(G)$, M_{2j} is a dominating set of vertices in G.*

Proof. With no loss of generality, we assume $|I_0| \leq |I_1| \leq |I_2| \leq \cdots \leq |I_{\Delta(G)}|$.

Proof of (1). By the monotonicity of the cardinality of the islands, it suffices to show that $|I_{\Delta(G)}| = 1$. Suppose to the contrary that $|I_{\Delta(G)}| \geq 2$. Then $\sum_{i=1}^{\Delta(G)} |C(I_i)| \geq 2 + \sum_{i=1}^{\Delta(G)-1} |C(I_i)| \geq 2 + \sum_{i=1}^{\Delta(G)-1} 1 \geq \Delta(G)+1$, contradicting Lemma 3.3. Since each island is thus an atoll and no two holes are consecutive [see Lemma 2.2 in [12]], then $I_j = \{2j\}$ for $0 \leq j \leq \Delta(G)$.

Proof of (2). From (1), $\mathrm{rc}(I_{\Delta(G)}) = 2\Delta(G)$. But $\lambda(G) = \mathrm{rc}(I_{\rho(G)}) = \mathrm{rc}(I_{\Delta(G)})$, since $\Delta(G) = \rho(G)$.

Proof of (3). Since each vertex of G is assigned a coastal label under L, the result follows from Lemma 3.1.

Proof of (4). For each fixed j, $0 \leq j \leq \Delta(G)$, and each $i \neq j$, $0 \leq i \leq \Delta(G)$, each vertex in M_{2i} is adjacent to some vertex in M_{2j} by Lemma 3.1. □

We note that in the next section, additional consideration will be given to the structure of graphs in the case $\Delta(G) = \rho(G)$.

THEOREM 3.6. *Let G be a graph with $\Delta(G) \geq 1$ and $\rho(G) = \Delta(G) - 1$. Then $2\Delta(G) - 1 \leq \lambda(G) \leq 2\Delta(G)$. Furthermore, if $\lambda(G) = 2\Delta(G)$, then*

1. *If $\Delta(G) = 1$, then $G = mK_2 + nK_1$ where $m, n \geq 1$.*
2. *If $\Delta(G) = 2$, then $G = nC_4$ or $H + K_1$ where H is a graph with $\rho(H) = \Delta(G)$.*
3. *If $\Delta(G) \geq 3$, then $G = H + K_1$ where H is a graph with $\rho(H) = \Delta(G)$.*

Proof. To show that $2\Delta(G) - 1 \leq \lambda(G)$, we note that since $K_{1,\Delta(G)}$ is a subgraph of G, every $L(2,1)$-labeling L uses at least $\Delta(G)+1$ distinct labels. Since $\rho(G) = \Delta(G) - 1$, it follows that $s(L) \geq 2\Delta(G) - 1$.

We next show that $\lambda(G) \leq 2\Delta(G)$. For any graph G, if $\Delta(G) = 1$ (resp. 2), then $\lambda(G) = 2$ (resp. ≤ 4). In the case $\Delta(G) \geq 3$, suppose to the contrary that $\lambda(G) \geq 2\Delta(G) + 1$, and let $L \in \Lambda_\rho(G)$ with $|I_0(L)| \leq |I_1(L)| \leq |I_2(L)| \leq \cdots \leq |I_{\Delta(G)-1}(L)|$. We observe that $|I_{\Delta(G)-2}(L)| = 1$; otherwise, $\sum_{i=1}^{\Delta(G)-1} |C(I_i(L))| \geq 4 + \sum_{i=1}^{\Delta(G)-3} |C(I_i(L))| \geq 4 + (\Delta(G) - 3) = \Delta(G) + 1$, contradicting Lemma 3.3. Since it follows that $I_j(L) = \{2j\}$ for $0 \leq j \leq \Delta(G) - 2$, then $I_{\Delta(G)-1}(L) = \{2\Delta(G) - 2, 2\Delta(G) - 1, \ldots, \lambda(G)\}$. But $\lambda(G) \geq 2\Delta(G) + 1$, implying that $|I_{\Delta(G)-1}(L)| \geq 4$ and hence $|C(I_{\Delta(G)-1}(L))| = 2$. Therefore, by the arbitrariness of L, every element of $\Lambda_\rho(G)$ induces $\Delta(G)$ islands, exactly $\Delta(G) - 1$ of which are atolls.

By Lemma 3.1, each vertex in $M_0(G,L)$ has degree $\Delta(G)$ and is thus adjacent only to vertices with labels in $\bigcup_{i=1}^{\Delta(G)-1} C(I_i(L))$. This implies that no vertex with label 0 is adjacent to a vertex with label $2\Delta(G) - 1$. It similarly follows that no vertex with label 2 is adjacent to any vertex with label $2\Delta(G) - 1$. Therefore, given fixed $v_0 \in M_{2\Delta(G)-1}(G,L)$, we may produce a new λ-labeling L' of G as follows:

$$L'(v) = \begin{cases} L(v) & \text{if } v \neq v_0, \\ 1 & \text{if } v = v_0. \end{cases}$$

If $m_{2\Delta(G)-1}(G,L) \geq 2$, then L' has $\Delta(G) - 2 < \rho(G)$ holes, a contradiction of the minimality of $\rho(G)$. If $m_{2\Delta(G)-1}(G,L) = 1$, then L' is in $\Lambda_\rho(G)$ and induces $\Delta(G)$ islands of which exactly $\Delta(G) - 2$ are atolls. But this contradicts the earlier observation that every element of $\Lambda_\rho(G)$ induces $\Delta(G)$ islands, exactly $\Delta(G) - 1$ of which are atolls. These contradictions imply that $\lambda(G) \leq 2\Delta(G)$.

We now turn to parts (1), (2), and (3). Suppose $\lambda(G) = 2\Delta(G)$, with $\rho(G) = 2\Delta(G) - 1$.

Proof of (1). Obvious.

Proof of (2). If $\Delta(G) = 2$, then $\lambda(G) = 4$ and $\rho(G) = 1$. If $L \in \Lambda_\rho(G)$, then L induces the following islands:

$I_0 = \{0\}$, $I_1 = \{2, 3, 4\}$, or
$I_0 = \{0, 1, 2\}$, $I_1 = \{4\}$, or
$I_0 = \{0, 1\}$, $I_1\{3, 4\}$.

In the first of these cases, every vertex in M_0 has degree 2, and, by Lemma 3.1, is adjacent to some vertex in M_2 and some vertex in M_4. Thus, no vertex in M_3 is adjacent to a vertex in M_0. Moreover, since no vertex in M_3 can be adjacent to a vertex in M_2 or M_4, then each vertex in M_3 is isolated. Now fix $v \in M_3$. If $m_3 \geq 2$, then we can produce a new λ-labeling L' of G with no holes by relabeling v with 1, contradicting the minimality of $\rho(G)$. Therefore $m_3 = 1$, whence $G = H + K_1$, where $H = (V(G) - \{v\}, E(G))$ is a graph with $\rho(H) = \Delta(G) = 2$. A similar argument can be applied to the case $I_0 = \{0, 1, 2\}$, $I_1 = \{4\}$.

If $I_0 = \{0, 1\}$ and $I_1 = \{3, 4\}$, then

1. every vertex in M_0 is adjacent to some vertex in M_3 and some vertex in M_4;
2. every vertex in M_1 is adjacent to some vertex in M_3 and some vertex in M_4;
3. every vertex in M_3 is adjacent to some vertex in M_0 and some vertex in M_1; and
4. every vertex in M_4 is adjacent to some vertex in M_0 and some vertex in M_1.

Thus, G is a 2-regular graph and hence is a sum of cycles. Furthermore, since L has a hole at two, each cycle of G has length $4k$, $k \geq 1$. However, for any $k \geq 2$, it can be easily shown that a cycle of length $4k$ has a λ-labeling with no holes. Thus $k = 1$.

Proof of (3). Suppose $\Delta(G) \geq 3$, $\rho(G) = \Delta(G) - 1$, and $\lambda(G) = 2\Delta(G)$. Let $L \in \Lambda_\rho(G)$ with $|I_0| \leq |I_1| \leq |I_2| \leq \cdots \leq |I_{\Delta(G)-1}|$. Since $\lambda(G) = 2\Delta(G)$ and $\mathrm{lc}(I_{\Delta(G)-2}) \geq 2\Delta(G) - 4$, either $|I_{\Delta(G)-2}| = |I_{\Delta(G)-1}| = 2$ or $|I_{\Delta(G)-2}| = 1$ and $|I_{\Delta(G)-1}| = 3$. In the former case, each vertex in M_0 has degree $\Delta(G) + 1$ by Lemma 3.1, a contradiction. In the latter case, I_j is an atoll for $0 \leq j \leq \Delta(G) - 2$, and $I_{\Delta(G)-1} = \{2\Delta(G) - 2, 2\Delta(G) - 1, 2\Delta(G)\}$. Therefore, by arguments identical to those given for the first case of (2), $M_{2\Delta(G)-1}$ contains exactly one vertex v, and that vertex is isolated. Thus $G = H + K_1$ where $H = (V(G) - \{v\}, E(G))$ is a graph with $\rho(H) = \Delta(G)$. □

Theorem 3.7. *For arbitrary $k \geq 1$, there is no k-regular graph G with $\rho(G) = k - 1$ except for $k = 2$ and $G = nC_4$, $n \geq 1$.*

Proof. Suppose $k \geq 3$ and let G be k-regular with $\rho(G) = k - 1$. By Theorem 3.6(3), $\lambda(G) = 2k - 1$ since G has no isolated vertex. Let $L \in \Lambda_\rho(G)$ with $|I_0| \leq |I_1| \leq |I_2| \leq \cdots \leq |I_{k-1}|$. Then $I_j = \{2j\}$ for $0 \leq j \leq k - 2$ and $I_{k-1} = \{2k - 2, 2k - 1\}$. Let $v \in M_{2k-2}$. Then v can be adjacent only to vertices with labels in I_j, $0 \leq j \leq k - 2$, implying $d(v) = k - 1$, a contradiction to the k-regularity of G. The cases $k = 1, 2$ follow from inspection. □

Corollary 3.8. *Let G be a graph with $\delta(G) \geq 1$. If $\delta(G) \leq \Delta(G) - 2$, then $\rho(G) \leq \Delta(G) - 2$.*

Proof. By Theorem 3.4, $\rho(G) \leq \Delta(G)$. If $\rho(G) = \Delta(G)$, then by Theorem 3.5, G is $\Delta(G)$-regular, and $\delta(G) = \Delta(G)$. So, suppose $\rho(G) = \Delta(G) - 1$. Then by Theorem 3.6, $\lambda(G) \leq 2\Delta(G)$. If $\lambda(G) = 2\Delta(G)$, then by Theorem 3.6 and the assumption $\delta(G) \geq 1$, it follows that $\Delta(G) = 2$, implying the contradiction $\delta(G) - 2 \leq 0$. Therefore $\lambda(G) = 2\Delta(G) - 1$. Arguing as above, let $L \in \Lambda_\rho(G)$ with $|I_0| \leq |I_1| \leq |I_2| \leq \cdots \leq |I_{\Delta(G)-1}|$. Then every island under L is necessarily an atoll except $I_{\Delta(G)-1} = \{2\Delta(G) - 2, 2\Delta(G) - 1\}$. So, for $v \in M_{2\Delta(G)-2}$, $d(v) = \Delta(G) - 1$ by Lemma 3.1, a contradiction to the assumption $\delta(G) \leq \Delta(G) - 2$. □

THEOREM 3.9. *Let G be k-regular and let $L \in \Lambda_\rho(G)$ with $|I_0| \leq |I_1| \leq \cdots \leq |I_{\rho(G)}|$. Then*

1. *If $|I_{\rho(G)}| = 1$, then $\rho(G) = k$ and $\lambda(G) = 2k$.*
2. *If $|I_{\rho(G)}| = 2$, then $\rho(G) \geq 1$, $|I_j| = 2$ for all $0 \leq j \leq \rho(G)$, $k = 2\rho(G)$, and $\lambda(G) = 3\rho(G) + 1 = \frac{3}{2}k + 1$.*
3. *If $|I_{\rho(G)}| \geq 3$, then $k \geq 2$, $\rho(G) \leq k - 2$, and $\lambda(G) \geq k + 2 + \rho(G)$.*

Proof. (1) There are $\rho(G)+1$ islands of L, each of which is an atoll since $|I_{\rho(G)}| = 1$. Thus, by Lemma 3.1, $k = \rho(G)$, from which it follows from Theorem 3.5 that $\lambda(G) = 2k$.

Proof of (2). If $\rho(G) = 0$, then $I_0 = \{0,1\}$, implying the contradiction that $\lambda(G) = 1$. So $\rho(G) \geq 1$. We now show that $|I_j| = 2$ for all $0 \leq j \leq \rho(G)$ by showing that $|I_0| = 2$.

We observe that each island under L contains only coastal labels since $|I_{\rho(G)}| = 2$. Let w be a vertex with $L(w) \in I_{\rho(G)}$. Since G is k-regular, Lemma 3.1 implies that for every label $l \in I_j \neq I_{\rho(G)}$, w is adjacent to some vertex labeled l. Hence, $\sum_{i=0}^{\rho(G)-1} |I_i| = k$. By similar consideration of a vertex v with $L(v) \in I_0$, we have $\sum_{i=1}^{\rho(G)} |I_i| = k$. Thus, by the two summations, $|I_0| = |I_{\rho(G)}| = 2$.

Since $|I_j| = 2$ for all j, $0 \leq j \leq \rho(G)$, we have $I_j = \{3j, 3j+1\}$. Hence, $\lambda(G) = 3\rho(G) + 1$. But as indicated above, for v a vertex with $L(v) = 0$, v has neighbors with labels precisely the elements of $\bigcup_{i=1}^{\rho(G)} I_i$. Hence, $k = |\bigcup_{i=1}^{\rho(G)} I_i| = 2\rho(G)$, so $\lambda(G) = \frac{3}{2}k + 1$.

Proof of (3). Since $I_{\rho(G)} \geq 3$, the label $\lambda(G) - 1$ is an interior label. Thus, for vertex v with $L(v) = \lambda(G) - 1$, the neighbors of v are assigned distinct labels not in $\{\lambda(G) - 2, \lambda(G) - 1, \lambda(G)\}$, implying that L assigns at least $d(v) + 3 = k + 3$ labels. Hence, $\lambda(G) \geq (k + 3 + \rho(G)) - 1 = k + 2 + \rho(G)$.

To show that $\rho(G) \leq k - 2$, we note by Theorem 3.4 that $\rho(G) \leq k$. Since not every island of L is an atoll, then $\rho(G) \neq k$ by Theorem 3.5. The result follows by Theorem 3.7 and the observation that $|I_{\rho(G)}| \geq 3$ implies that G cannot be a sum of 4-cycles. □

We note that K_n and the complete multipartite graphs $K_{2,2,\ldots,2}$ satisfy Theorem 3.9(1) and (2), respectively. In regard to Theorem 3.9(3), the bound $k + 2 + \rho(G)$ is not necessarily sharp. For example, we argue as follows that there is no 5-regular graph G such that $\rho(G) = 3$ and $\lambda(G) = 10$. Suppose to the contrary that such a graph exists. Let $L \in \Lambda_\rho(G)$ such that $|I_0| \leq |I_1| \leq |I_2| \leq |I_3|$. Then $|I_0| \leq 2$ (for otherwise $\lambda(G) > 10$). If $|I_0| = 2$, then $I_0 = \{0,1\}, I_1 = \{3,4\}, I_2 = \{6,7\}$ and $I_3 = \{9,10\}$. Hence, by Lemma 3.1, each vertex v has degree 6, a contradiction. Thus, $|I_0| = 1$. Noting that $|I_1| \leq 2$, if $|I_1| = 2$, then $I_0 = \{0\}, I_1 = \{2,3\}, I_2 = \{5,6\}$ and $I_3 = \{8,9,10\}$. Hence, by Lemma 3.1, each vertex v with $L(v) = 0$ has degree 6, another contradiction. Thus $|I_1| = 1$. Now, $|I_2|$ is either 1, 2, or 3. If $|I_2| = 3$, then $I_0 = \{0\}, I_1 = \{2\}, I_2 = \{4,5,6\}$, and $I_3 = \{8,9,10\}$. Thus, by Lemma 3.1, each vertex v with $L(v) = 0$ has neighbors with labels 2, 4, 6, 8, and 10. But by the distance 1 condition and the 5-regularity of G, each vertex w with $L(w) = 9$ has a neighbor with label 0, a contradiction. A similar argument which focuses on vertices with labels 0 and 8 demonstrates that $|I_2|$ cannot be 2. Hence, $|I_2| = 1$. In this case, we have $I_0 = \{0\}, I_1 = \{2\}, I_2 = \{4\}$, and $I_3 = \{6,7,8,9,10\}$. So, by Lemma 3.1 and the 5-regularity of G, each vertex v with $L(v) \neq 0$ has a neighbor labeled 0. Thus, M_0 is a dominating set, and $|V(G)| = 6m_0$ (since G is 5-regular). Since $m_0 = m_{10}$ by Lemma 3.1, M_{10} is a dominating set as well. Therefore, since $M_9 \neq \phi$, there are adjacent vertices with respective labels 9 and 10, a contradiction.

We have been unable to find a 5-regular graph G with $\rho(G) = 3$. We conjecture that if G is a k-regular graph with $\rho(G) \geq 1$, then $\rho(G)$ divides k.

4. Relating $\rho(G)$, $\lambda(G)$, and $\mu(G)$. For purposes of this discussion, it will be convenient to consider the two cases $\lambda(G) \geq n-1$ and $\lambda(G) \leq n-2$, where $n = |V(G)|$. We begin with the case $\lambda(G) \geq n-1$.

THEOREM 4.1. *Let G be a graph with order n and $\lambda(G) \geq n-1$. Then*

1. $\rho(G) = c(G^c) - 1 = \lambda(G) - (n-1)$, *and*
2. *for $L \in \Lambda_\rho(G)$, $m_i(G, L) = 0$ or 1.*

Proof of (1). Since $\lambda(G) \geq n-1$, it follows from Theorem 2.5 that $c(G^c) - 1 = \lambda(G) - (n-1)$.

Let $\mathcal{C}$ be a path covering of G^c with minimum order. Then $\mathcal{C}$ induces a λ-labeling of G with $c(G^c) - 1$ holes (see [12]). Hence, $\rho(G) \leq c(G^c) - 1 = \lambda(G) - (n-1)$.

Now let $L \in \Lambda_\rho(G)$ and let $H(L)$ and $N(L)$ denote the set of holes of L and the set of labels assigned by L, respectively. We observe that $|H(L)| = \rho(G)$ and that $|H(L)|+|N(L)|-1 = \lambda(G)$. Thus, $\lambda(G) = (n-1)+(c(G^c)-1) = |H(L)|+|N(L)|-1 = \rho(G) + |N(L)| - 1 \leq \rho(G) + n - 1$, giving $\rho(G) \geq \lambda(G) - (n-1)$.

Proof of (2). Select $L \in \Lambda_\rho(G)$. We have seen $\lambda(G) = n + c(G^c) - 2 = |N(L)| + \rho(G) - 1$. It thus follows that $n = |N(L)|$ by (1). □

COROLLARY 4.2. *Let G be a graph with order n and $\lambda(G) \geq n-1$. Then*

1. $c(G^c) \leq \Delta(G) + 1$, *and*
2. $\rho(G) \leq \chi(G) - 1$.

Proof.

1. By Theorems 4.1 and 3.4, $c(G^c) - 1 = \rho(G) \leq \Delta(G)$.
2. For any graph G, $c(G^c) \leq \chi(G)$. The result follows by Theorem 4.1. □

We now turn our attention to graphs G with $\lambda(G) \leq n-2$, and consider the upper bound on the invariant $\mu(G)$ given by Fishburn and Roberts in the following theorem.

THEOREM 4.3. *See* 7. *If G is a graph such that $\rho(G) \geq 1$ and $\lambda(G) \leq n-2$, then $\mu(G) \leq \lambda(G) + \rho(G)$.*

It is easily seen that for $\rho(G) \geq 1$, a lower bound for $\mu(G)$ is $\lambda(G) + 1$. Thus by Theorem 4.3, $\mu(G) = \lambda(G) + 1$ if $\rho(G) = 1$. It is also immediate from Theorem 3.4 that an alternative upper bound for $\mu(G)$ is $\lambda(G) + \Delta(G)$.

We now improve the upper bound of $\lambda(G) + \rho(G)$ in the cases $\rho(G) = \Delta(G) - 1$ and $\rho(G) = \Delta(G)$.

THEOREM 4.4. *Suppose G is a graph with order n, $\lambda(G) \leq n-2$, and $\rho(G) = \Delta(G) \geq 1$. Then $\mu(G) = \lambda(G) + 1$.*

Proof. By Theorem 3.5, G is Δ-regular with $\lambda(G) = 2\Delta$, and for each L in $\Lambda_\rho(G)$, L induces $\Delta + 1$ islands $I_0, I_1, \ldots, I_\Delta$, where I_i is the atoll $\{2i\}$. By Lemma 3.1 and Theorem 3.5(3), then $n = m_0(\Delta+1)$, implying $2\Delta \leq m_0(\Delta+1) - 2$. This gives $m_0 \geq 2$.

By Lemma 3.1, we may denote the m_0 elements of M_{2i} by $v_{1,2i}, v_{2,2i}, \ldots, v_{m_0,2i}$ where, with no loss of generality, $v_{j,2i}$ is adjacent to $v_{j,2i+2}$. In particular, with j fixed equal to 1, $v_{1,0}, v_{1,2}, v_{1,4}, \ldots, v_{1,2\Delta}$ is a path in G. It now suffices to produce a no-hole $L(2,1)$-labeling L^* of G with span $2\Delta + 1 = \lambda(G) + 1$, which we do as follows:

$$L^*(v) = \begin{cases} L(v) + 1 & \text{if } v = v_{1,2i} \text{ for some } i, \\ L(v) & \text{otherwise.} \end{cases} \qquad \square$$

THEOREM 4.5. *Suppose G is a graph with order n, $\lambda(G) \leq n-2$, and $\rho(G) = \Delta(G) - 1$. Then*

1. $\mu(G) = \lambda(G)$ *if* $\Delta(G) = 1$;
2. $\mu(G) = \lambda(G) + 1$ *if* $\Delta(G) \geq 2$.

Proof. By Theorem 3.6, $2\Delta(G) - 1 \leq \lambda(G) \leq 2\Delta(G)$. We first consider the case $\lambda(G) = 2\Delta(G)$.

Case 1: $\lambda(G) = 2\Delta(G)$.

If $\Delta(G) = 1$, then $\rho(G) = 0$, implying $\mu(G) = \lambda(G)$.

If $\Delta(G) = 2$, then by Theorem 3.6(2), G is isomorphic to either mC_4 (for some positive integer m) or $H + K_1$ where $\rho(H) = \Delta(G) = 2$. In the former case, $\lambda(G) = 4 \leq n-2 = 4m-2$, implying $m \geq 2$. By labeling the vertices of $m-1$ copies of C_4 with integers 0, 3, 1, 4, and labeling the vertices in the remaining copy of C_4 with integers 1, 4, 2, 5, we produce a no-hole $L(2,1)$-labeling of H with span $5 = \lambda(G)+1$. Thus, there exists a no-hole labeling of G with span $\lambda(G) + 1$ as well. But $\rho(G) = \Delta(G) - 1 = 1$, so $\mu(G) > \lambda(G)$. This implies $\mu(G) = \lambda(G) + 1$. In the latter case, Fishburn and Roberts [6] show that H is necessarily isomorphic to $mC_3 + kC_6$ for some integers $m, k \geq 0$. Since $4 = \lambda(G) \leq n - 2 = (3m + 6k + 1) - 2$, it follows that $m \geq 2$ or $k \geq 1$. In either event, it is easy to establish a no-hole $L(2,1)$-labeling of H with span $5 = \lambda(G) + 1$, from which it follows as above that $\mu(G) = \lambda(G) + 1$.

If $\Delta(G) \geq 3$, then by Theorem 3.6(3), G is isomorphic $H + K_1$ where $\rho(H) = \Delta(G)$. But $\Delta(G) = \Delta(H)$, so by Theorem 3.5, $\lambda(H) = 2\Delta(H)$ and $|V(H)| = w(\Delta(H) + 1)$ for some $w \geq 1$. Hence, since $\lambda(H) = \lambda(G) \leq n - 2$, we have $2\Delta(H) \leq n - 2 = |V(H)| + 1 - 2 = w(\Delta(H) + 1) - 1$, implying $w \geq 2$. This implies $\lambda(H) \leq |V(H)| - 2$. By Theorem 4.4, $\mu(H) = \lambda(H) + 1 = \lambda(G) + 1$, which implies that H (and therefore G) have no-hole labelings with span $\lambda(G) + 1$. But $\rho(G) = \Delta(G) - 1 > 1$, so $\mu(G) > \lambda(G)$. Thus $\mu(G) = \lambda(G) + 1$.

We now turn to the case $\lambda(G) = 2\Delta(G) - 1$. Let $L \in \Lambda_\rho(G)$, where $|I_0| \leq |I_1| \leq \cdots \leq |I_\rho|$. Then $I_j = \{2j\}$ for $0 \leq j \leq \rho - 1$ and $I_\rho = \{2\rho, 2\rho + 1\} = \{2\Delta(G) - 2, 2\Delta(G) - 1\}$. Hence, L assigns $\rho(G) + 2 = \Delta(G) + 1$ distinct labels, each of which is coastal. By Lemma 3.1, $m_i = m_0$ for every label i assigned by L. Therefore $n = m_0(\Delta(G)+1)$, giving $\lambda(G) = 2\Delta(G)-1 \leq n-2 = m_0(\Delta(G)+1)-2$, which implies $m_0 \geq 2$. For $0 \leq i \leq \Delta(G)-1$, let $M_{2i} = \{v_{1,2i}, v_{2,2i}, \ldots, v_{m_0,2i}\}$. By Lemma 3.1 and without loss of generality, we may suppose $v_{j,2i}$ is adjacent to $v_{j,2i+2}$, $1 \leq j \leq m_0$, $0 \leq i \leq \Delta(G) - 2$. In particular, with j fixed equal to 1, $v_{1,0}, v_{1,2}, v_{1,4}, \ldots, v_{1,2\Delta(G)-2}$ is a path in G. It now suffices to produce a no-hole $L(2,1)$-labeling L^* of G with span $\lambda(G) + 1 = 2\Delta(G)$, which we perform as follows:

$$L^*(v) = \begin{cases} L(v) & \text{if } v = v_{1,2i} \text{ for some } i, 0 \leq i \leq \Delta(G) - 1, \\ L(v) + 1 & \text{otherwise.} \end{cases} \qquad \Box$$

5. On the structure of graphs G with $\rho(G) = \Delta(G)$. As shown in Theorem 3.5, for each graph G with $\rho(G) = \Delta(G)$ and each $L \in \Lambda_\rho(G)$,

1. G is Δ-regular with $|V(G)| \equiv 0$ *mod* $(\Delta(G) + 1)$;
2. $\lambda(G) = 2\Delta(G)$;
3. $M_{2j}(G, L)$ is a dominating set for each j, $0 \leq j \leq \Delta(G)$;
4. $I_j = \{2j\}$ for each j, $0 \leq j \leq \Delta(G)$.

Let $\mathcal{G}_{\Delta,t}$ be the collection of connected graphs G with $\rho(G) = \Delta(G) = \Delta$ and order $t(\Delta + 1)$ (implying $m_{2j}(G, L) = t$ for every $L \in \Lambda_\rho(G)$ and each j, $0 \leq j \leq \Delta(G)$.) Let $\mathcal{B}_{\Delta,t}$ be the subcollection of graphs in $\mathcal{G}_{\Delta,t}$ which are bipartite. We note that $\mathcal{G}_{\Delta,1} = \{K_{\Delta+1}\}$. We thus restrict our attention to the case $t \geq 2$, with particular emphasis on $t = 2$.

In [7], Fishburn and Roberts construct connected graphs G with $\lambda(G) = 2m$, $|V(G)| = 2(m+1)$, and $\rho(G) = m$, for $m \geq 2$. We note that for $m = 2$, the constructed graph is isomorphic to C_6, and for $m \geq 3$, the constructed graph is not bipartite. Thus, it follows that for $\Delta \geq 2$, $\mathcal{B}_{2,2}$, and $\mathcal{G}_{\Delta,2}$ are not empty. We also note that $\mathcal{B}_{2,2} = \mathcal{G}_{2,2}$.

The following lemma will assist in characterizing $\mathcal{B}_{\Delta,2}$ for all $\Delta \geq 2$.

Lemma 5.1. *If G is a connected Δ-regular graph of order $2(\Delta+1)$, then $G \in \mathcal{G}_{\Delta,2}$ or $\lambda(G) = 2\Delta + 1$.*

Proof. Since G^c is a $(\Delta+1)$-regular graph on $2(\Delta+1)$ vertices, then by Dirac's theorem [5], G^c has a Hamilton path. Hence, by Theorem 2.5, $\lambda(G) \leq |V(G)| - 1 = 2\Delta + 1$. It suffices to show that if $\lambda(G) \leq 2\Delta$, then $G \in \mathcal{G}_{\Delta,2}$.

Let L be an arbitrary $L(2,1)$-labeling of G with span $s(L)$, $\lambda(G) \leq s(L) \leq 2\Delta$. If v and w are vertices in $V(G)$ such that $L(v) = L(w) = l$, then $\{v, w\}$ is a dominating set due to the distance conditions and regularity and order of G. Hence, there exists no vertex with label $l-1$ or $l+1$, which in turn implies $m_i + m_{i+1} \leq 2$ for each i, $0 \leq i \leq s(L) - 1$. Therefore, $|V(G)| = 2\Delta + 2 \leq 2\lfloor \frac{s(L)+2}{2} \rfloor$, giving $s(L) \geq 2\Delta$. Since L was arbitrary, $\lambda(G) \geq 2\Delta$ as well, giving $\lambda(G) = 2\Delta$.

Now let L be an arbitrary λ-labeling of G. To see that L necessarily has Δ holes, we note that since $\lambda(G) = 2\Delta$, then $|V(G)| = 2\Delta+2 = (m_0+m_1)+(m_2+m_3)+\cdots+(m_{2\Delta-2}+m_{2\Delta-1})+m_{2\Delta} = m_0+(m_1+m_2)+(m_3+m_4)+\cdots+(m_{2\Delta-1}+m_{2\Delta})$. Since $m_i+m_{i+1} \leq 2$ as above, then $m_i+m_{i+1} = 2$ for all i, $0 \leq i \leq 2\Delta-1$, and $m_0, m_\Delta = 2$ as well. Hence, for $0 \leq i \leq 2\Delta - 2$, $(m_{i+2} + m_{i+1}) - (m_{i+1} + m_i) = m_{i+2} - m_i = 0$, which gives $m_i = 2$ for even i and $m_i = 0$ for odd i. □

Now, for $\Delta \geq 2$, let ${}_{\triangle}B$ be a connected Δ-regular bipartite graph with order $2(\Delta+1)$. It is easy to see that ${}_{\triangle}B$ can be obtained by deleting a perfect matching from $K_{\Delta+1,\Delta+1}$, and is unique up to isomorphism.

Theorem 5.2. *For $\Delta \geq 2$, $\mathcal{B}_{\Delta,2} = \{{}_{\triangle}B\}$.*

Proof. Since ${}_{\triangle}B$ has diameter 3, then for every vertex $v \in V({}_{\triangle}B)$, there exists a unique vertex $w \in V({}_{\triangle}B)$ such that $d(v,w) = 3$. Hence there exists an $L(2,1)$-labeling of ${}_{\triangle}B$ with span 2Δ. Thus, by Lemma 5.1, ${}_{\triangle}B \in \mathcal{G}_{\Delta,2}$, implying ${}_{\triangle}B \in \mathcal{B}_{\Delta,2}$. □

From Theorem 5.2 and the discussion preceding Lemma 5.1, it follows that $|\mathcal{G}_{m,2}| \geq 2$ for $m \geq 3$. We further note that $\mathcal{B}_{3,2} = \{Q_3\}$.

To determine $\mathcal{G}_{3,2}$, we consider the four nonisomorphic connected 3-regular graphs of order 8 (see [1]) as shown in Figure 5.1.

The graph in Figure 5.1(a) is the graph constructed by Fishburn and Roberts, while the graph in Figure 5.1(b) is Q_3. Each is clearly in $\mathcal{G}_{3,2}$. On the other hand, if $G \in \mathcal{G}_{\Delta,2}$, then $V(G)$ can be partitioned into $\Delta(G) + 1$ sets containing precisely 2 vertices which are exactly distance 3 apart. Since the diameter of the graph in Figure 5.1(d) is 2, its λ-number is 7 by Lemma 5.1. And since, in Figure 5.1(c), there is a vertex which is at most distance 2 from every other vertex, that graph is not in $\mathcal{G}_{3,2}$. It follows from Lemma 5.1 that the λ-number of this graph is 7 as well.

We next introduce a particular graph construction which will aid in characterizing $\mathcal{G}_{\Delta,2}$.

5.1. The S-exchange of the sum of two graphs. Let G be a graph with $V(G) = \{v_0, v_1, v_2, \ldots, v_{n-1}\}$ and for $i = 1, 2$, let ϕ_i be a graph isomorphism from G to graph G_i where $\phi_i(v_j) = v_{j,i}$. Let $e = \{v_r, v_s\} \in E(G)$. Then the e-exchange of graph $G_1 + G_2$, denoted $X_e(G_1 + G_2)$, is the graph with vertex set $V(G_1 + G_2)$ and edge set

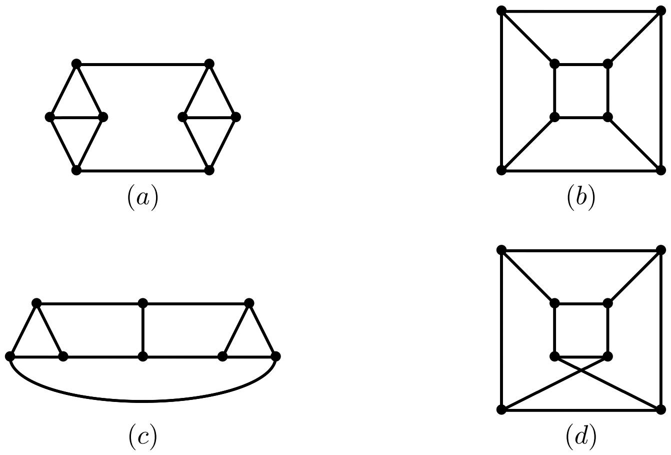


FIG. 5.1. *Four nonisomorphic connected* 3*-regular graphs of order* 8.

$(E(G_1+G_2)-\{\phi_1(e),\phi_2(e)\})\bigcup T(e)$, where $T(e)=\{\{v_{r,1},v_{s,2}\},\{v_{r,2},v_{s,1}\}\}$. Furthermore, if $S\subseteq E(G)$, then the S-exchange of graph G_1+G_2, denoted $X_S(G_1+G_2)$, is the graph with vertex set $V(G_1+G_2)$ and edge set $(E(G_1+G_2)-\bigcup_{e\in S}\{\phi_1(e),\phi_2(e)\})\bigcup(\bigcup_{e\in S}T(e))$.

By way of illustration, we note that if G is isomorphic to K_3 and $S=E(G)$, then $X_S(G_1+G_2)$ is isomorphic to C_6. Additionally, if G is isomorphic to K_4 and e is any edge in $E(G)$, then $X_e(G_1+G_2)$ is isomorphic to the graph in Figure 5.1(a). We also note that for any $v\in V(G)$, if $S(v)=\{e\in E(G)|e$ is incident to $v\}$, then $X_{S(v)}(G_1+G_2)$ is isomorphic to G_1+G_2.

THEOREM 5.3. *Let H be a connected Δ-regular graph with order $2(\Delta+1)$. Then $H\in\mathcal{G}_{\Delta,2}$ if and only if there exists $S\subseteq E(K_{\Delta+1})$ such that H is isomorphic to $X_S(K_{\Delta+1}+K_{\Delta+1})$.*

Proof. ($\Rightarrow$). Let $H\in\mathcal{G}_{\Delta,2}$ and let L be a λ-labeling of H. Then for $0\le i\le 2\Delta$, $m_i=0$ if i is odd and $m_i=2$ if i is even. Let $v_{0,1}$ and $v_{0,2}$ denote the two vertices in $V(H)$ with label 0 under L. For $i=1,2$ and for $1\le j\le\Delta$, let $v_{j,i}$ be the vertex in $V(H)$ which has label $2j$ and which is adjacent to $v_{0,i}$. Also let H_i be the subgraph of H induced by $\{v_{0,i},v_{1,i},\ldots,v_{\Delta,i}\}$ and let W be the edge set of H_1^c. (We note that H_1 is isomorphic to H_2.) Setting $S=\phi_1^{-1}(W)$ (where ϕ_1 is the graph isomorphism from G to G_1 such that $\phi(v_i)=v_{i,1}$, where $G=K_{\Delta+1}$ and $V(G)=\{v_0,v_1,\ldots,v_\Delta\}$), we easily see that H is isomorphic to $X_S(K_{\Delta+1}+K_{\Delta+1})$.

($\Leftarrow$). Suppose $S\subseteq E(K_{\Delta+1})$ such that H is isomorphic to $X_S(K_{\Delta+1}+K_{\Delta+1})$. Let L be the $L(2,1)$-labeling of H such that $L(v_{j,i})=2j$ for $i=1,2$. Since the span of L is $2\Delta<2\Delta+1$, Lemma 5.1 implies that $H\in\mathcal{G}_{\Delta,2}$. □

It is easily seen that the graphs in Figures 5.1(a) and 5.1(b) are S-exchanges of K_4+K_4, where, in the latter case, $|S|=2$ (for independent edges) and in the former case, $|S|=1$.

To this point, we have restricted our attention to elements of $\mathcal{G}_{\Delta,t}$ for $t=2$. Using two new graph constructions, we next extend the discussion to $2<t\le\Delta(G)$.

The graph $\boldsymbol{\Omega_r}$. For $r\ge1$, let $X=rK_r$ and $Y=rK_1$. We form a new graph Ω_r by joining the vertices of Y to certain vertices of X. Formally, let $V(\Omega_r)=V(X)\bigcup V(Y)$ where

1. $V(X)=\bigcup_{i=0}^{r-1}B_i$, $B_i=\{b_{i,j}|0\le j\le r-1\}$, and

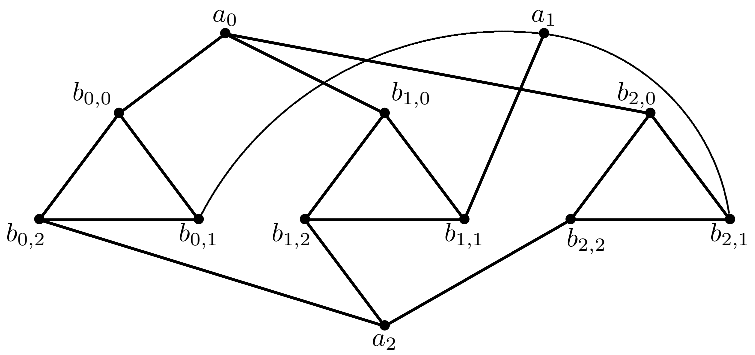


FIG. 5.2. *The graph* Ω_3.

2. $V(Y) = \{a_0, a_1, ..., a_{r-1}\}$.

Let $E(\Omega_r) = R \bigcup S$, where

3. $R = \bigcup_{i=0}^{r-1} R_i$, where $R_i = \{\{b_{i,j}, b_{i,k}\} | 0 \leq j < k \leq r-1\}$, and
4. $S = \bigcup_{i=0}^{r-1} S_i$, where $S_i = \{\{a_i, b_{m,i}\} | 0 \leq m \leq r-1\}$.

We note that Ω_1 is isomorphic to K_2, and Ω_2 is isomorphic to C_6. We illustrate Ω_3 in Figure 5.2.

We make the following observations about the structure of Ω_r:

Obs. 1) Ω_r is r-regular and has order $r^2 + r$; $|V(X)| = r^2$ and $|V(Y)| = r$;

Obs. 2) for $0 \leq i, j \leq r-1$, $d(a_j, a_i) = 3$ for $j \neq i$;

Obs. 3) for $0 \leq i, j, k, l \leq r-1$

$$d(b_{i,j}, b_{k,l}) = \begin{cases} 1 & \text{if } i = k \text{ and } j \neq l, \\ 2 & \text{if } i \neq k \text{ and } j = l, \\ 3 & \text{otherwise;} \end{cases}$$

Obs. 4) For $0 \leq i, j, k \leq r-1$,

$$d(a_i, b_{j,k}) = \begin{cases} 1 & \text{if } i = k, \\ 2 & \text{otherwise.} \end{cases}$$

LEMMA 5.4. *Let L be an $L(2,1)$-labeling of Ω_r. Then*

1. *for every $y \in V(Y)$ and every $x \in V(X)$, $L(x) \neq L(y)$;*
2. *for $0 \leq t \leq s(L) - 1$, $m_t + m_{t+1} \leq r$.*

Proof. By Obs. 4, (1) follows.

To show (2), suppose to the contrary that there exists t, $0 \leq t \leq s(L) - 1$, such that $m_t + m_{t+1} \geq r + 1$. From Obs. 2, 3, 4, either every vertex labeled t (resp. $t+1$) is in $V(X)$ or every vertex labeled t (resp. $t+1$) is in $V(Y)$. Furthermore, if every vertex in $M_t \bigcup M_{t+1}$ is in $V(Y)$, then we have the contradiction that $r + 1 \leq m_t + m_{t+1} \leq |V(Y)| = r$. Similarly, if every vertex in $M_t \bigcup M_{t+1}$ is in $V(X)$, then by the pigeon-hole principle, there exist two vertices $b_{i,j}, b_{k,l}$ in $M_t \bigcup M_{t+1}$ where $i = k$. Thus, $b_{i,j}$ and $b_{k,l}$ are adjacent, a contradiction of the assumption that their labels under L differ by at most 1. We have therefore established that either $M_t \subseteq V(Y)$ and $M_{t+1} \subseteq V(X)$ or $M_t \subseteq V(X)$ and $M_{t+1} \subseteq V(Y)$.

Suppose the former. Let $s_t = \{i | a_i \in M_t\}$ and let $s_{t+1} = \{k | b_{j,k} \in M_{t+1}$ for some $j\}$. We observe that $|s_t| = m_t$, and from Obs. 3, $|s_{t+1}| = m_{t+1}$. Noting that s_t and s_{t+1} are subsets of $\{0, 1, 2, \ldots, r-1\}$, $|s_t| + |s_{t+1}| = m_t + m_{t+1} \geq r + 1$ implies

$s_t \bigcap s_{t+1} \neq \phi$. Thus, for some integers y, z, $0 \leq y, z \leq r-1$, there exist adjacent vertices a_y and $b_{z,y}$ in $M_t \bigcup M_{t+1}$, a contradiction of the distance one condition on L.

A similar argument can be made in the latter case. □

THEOREM 5.5. *For $r \geq 1$, $\Omega_r \in \mathcal{G}_{r,r}$.*

Proof. We first establish that $\lambda(\Omega_r) = 2r$. Suppose $\lambda(\Omega_r) < 2r$. Let L be an $L(2,1)$-labeling of Ω_r with span $2r-1$. By Obs. 1, $r^2 + r = |V(\Omega_r)| = \sum_{i=0}^{2r-1} m_i$. However, by Lemma 5.4, $\sum_{i=0}^{2r-1} m_i = \sum_{j=0}^{r-1}(m_{2j} + m_{2j+1}) \leq r^2$, a contradiction. Hence, $\lambda(\Omega_r) \geq 2r$. To show that $\lambda(\Omega_r) = 2r$, let $B_k = \{b_{i,j} | (j-i) \equiv k \bmod r\}$, $0 \leq k \leq r-1$. Noting that $|B_k| = r$ and that vertices in B_k are pairwise distance 3 apart, we produce an $L(2,1)$-labeling L of Ω_r as follows:

$$L(v) = \begin{cases} 2k & \text{if } v \in B_k, \\ 2r & \text{otherwise.} \end{cases}$$

To show $\rho(\Omega_r) = r$, let L^* be any λ-labeling of Ω_r. Then $r^2 + r = \sum_{i=0}^{2r} m_i = m_{2r} + \sum_{i=0}^{2r-1} m_i$. By Lemma 5.4, $\sum_{i=0}^{2r-1} m_i \leq r^2$, implying $m_{2r} = r$. By Obs. 1 (the r-regularity of Ω_r in particular), M_{2r} is therefore a dominating set. Thus, $m_{2r-1} = 0$. Proceeding by induction, it is easily seen that for $0 \leq j \leq 2r$,

$$m_j = \begin{cases} r & \text{if } j \text{ is even,} \\ 0 & \text{if } j \text{ is odd.} \end{cases}$$

Hence, $\rho(\Omega_r) = r$. □

Theorem 5.5 establishes the fact that $\mathcal{G}_{r,r}$ is nonempty. Earlier discussions have demonstrated that $\mathcal{G}_{r,1} = \{K_{r+1}\}$, and that for $r \geq 2$, $\mathcal{G}_{r,2}$ is nonempty. The question is thus raised: for what values of t is $\mathcal{G}_{r,t}$ nonempty?

To see that such graphs exist for arbitrary $t < r$, we introduce one last graph construction.

The graph $\boldsymbol{\Omega_{r,t}}$. Fix integers t and r such that $1 \leq t \leq r$. Let $X = tK_r$ and let $Y = tK_1$. We form a new graph $\Omega_{r,t}$ by joining the vertices in Y to certain vertices in X. Formally, let $V(\Omega_{r,t})$ equal $V(X) \bigcup V(Y)$, where

1. $V(X) = \bigcup_{i=0}^{t-1} B_i$, $B_i = \{b_{i,j} | 0 \leq j \leq r-1\}$, and
2. $V(Y) = \{a_0, a_1, ..., a_{t-1}\}$.

Let $E(\Omega_{r,t}) = R \bigcup S \bigcup T$, where

3. $R = \bigcup_{i=0}^{t-1} R_i$ where $R_i = \{\{b_{i,j}, b_{i,k}\} | 0 \leq j < k \leq r-1\}$, and
4. $S = \bigcup_{i=0}^{t-1} S_i$, where $S_i = \{\{a_i, b_{m,i}\} | 0 \leq m \leq t-1\}$, and
5. $T = \bigcup_{i=0}^{t-1} T_i$, where $T_i = \{\{a_i, b_{i,j}\} | t \leq j \leq r-1\}$.

We illustrate $\Omega_{4,2}$ in Figure 5.3.

We note that $\Omega_{2,1}$ is isomorphic to K_3, and in general $\Omega_{r,1}$ is isomorphic to K_{r+1}. We also note that $\Omega_r = \Omega_{r,r}$, and that $\Omega_{3,2}$ is isomorphic to the graph in Figure 5.1(a).

Arguments similar to those used in the analysis of Ω_r demonstrate that $\Omega_{r,t}$ is a graph G with $\rho(G) = r$ and $m_{2i}(G, L) = t$ for $L \in \Lambda_\rho(G)$.

We observe that the edges of $\Omega_{r,t}$ may be manipulated to produce other graphs G with $\rho(G) = r$ and $m_i(G, L) = t$ for $L \in \Lambda_\rho(G)$. Such a graph is illustrated in Figure 5.4 for $r = 4, t = 2$.

We point out that the graphs in Figures 5.3 and 5.4 can be constructed as S-exchanges of $K_5 + K_5$.

We have been unable to establish that $\mathcal{G}_{r,t}$ is nonempty for $t > r$, and conjecture that $\mathcal{G}_{r,t} = \phi$ for all $t > r$.

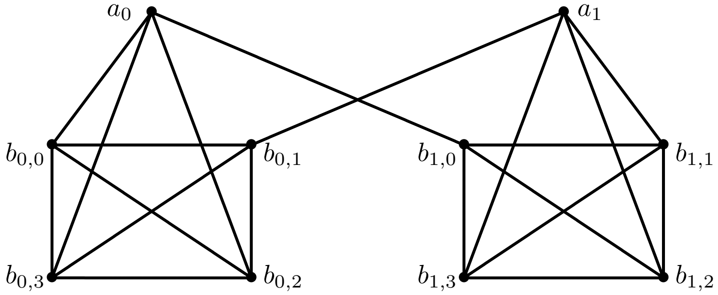


FIG. 5.3. *The graph* $\Omega_{4,2}$.

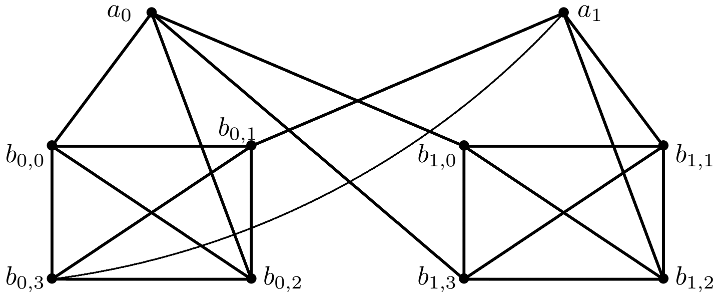


FIG. 5.4. *Graph* G *with* $\rho(G) = 4$ *and* $m_i(G, L) = 0$ *or* 2 *for* $L \in \Lambda(G, \rho)$.

6. Closing remarks. We have offered several conjectures about the structure of nonfull colorable graphs in earlier sections of this paper. Throughout our investigations of graphs G with positive $\rho(G)$ we found none with $\lambda(G) > 2\Delta(G)$. Thus, we conjecture that if $\lambda(G) > 2\Delta(G)$, then $\rho(G) = 0$.

Acknowledgement. The authors wish to thank the referees for their suggestions which greatly improved the paper.

REFERENCES

[1] F. C. BUSSEMAKER, S. COBELJIC, D. M. CVETKOVIC, AND J. J. SEIDEL, *Cubic graphs on* ≤ 14 *vertices*, J. Combinatorial Theory Ser. B, 23 (1977), pp. 234–235.

[2] G. J. CHANG, W.-T. KE, D. KUO, D. LIU, R. YEH, *On L(d,1)-labelings of graphs*, Discrete Math., 220 (2000), pp. 57–66.

[3] G. J. CHANG AND D. KUO, *The* $L(2,1)$*-labeling problem on graphs*, SIAM J. Discrete Math., 9 (1996), pp. 309–316.

[4] G. CHARTRAND, D. J. ERWIN, F. HARARY, AND P. ZHANG, *Radio labelings of graphs*, Bull. Inst. Combin. Appl., 33 (2001), pp. 77–85.

[5] G. A. DIRAC, *Some theorems on abstract graphs*, Proc. London. Math. Soc., 2 (1952), pp. 69–81.

[6] P. C. FISHBURN AND F. S. ROBERTS, *Full color theorems for* $L(2,1)$*-colorings*, preprint.

[7] P. C. FISHBURN AND F. S. ROBERTS, *No-hole* $L(2,1)$*-colorings*, Discrete Appl. Math., 130 (2003), pp. 513–519.

[8] J. P. GEORGES AND D. W. MAURO, *Generalized vertex labelings with a condition at distance two*, Congr. Numer., 109 (1995), pp. 141–159.

[9] J. P. GEORGES AND D. W. MAURO, *On the size of graphs labeled with a condition at distance two*, J. Graph Theory, 22 (1996), pp. 47–57.

[10] J. P. GEORGES AND D. W. MAURO, *Some results on* $\lambda_{j,k}$*-numbers of products of complete graphs*, Congr. Numer., 140 (1999), pp. 141–160.

[11] J. P. Georges, D. W. Mauro and M. I. Stein, *Labeling products of complete graphs with a condition at distance two*, SIAM J. Discrete Math., 14 (2001), pp. 28–35.

[12] J. P. Georges and D. W. Mauro, and M. A. Whittlesey, *Relating path coverings to vertex labellings with a condition at distance two*, Discrete Math., 135 (1994), pp. 103–111.

[13] J. R. Griggs and R. K. Yeh, *Labelling graphs with a condition at distance two*, SIAM J. Discrete Math., 5 (1992), pp. 586–595.

[14] W. K Hale, *Frequency assignment: Theory and application*, Proc. IEEE, 68 (1980), pp. 1497–1514.

[15] P. Jha, A. Narayanan, P. Sood, K. Sundaram, and V. Sunder, On $L(2,1)$*-labeling of the Cartesian product of a cycle and a path*, Ars Combin., 55 (2000), pp. 81–89.

[16] F. S. Roberts, T*-colorings of graphs: Recent results and open problems*, Discrete Math., 93 (1991), pp. 229–245.

[17] D. Sakai, *Labeling chordal graphs: Distance two condition*, SIAM J. Discrete Math., 7 (1994), pp. 133–140.

[18] J. Van Den Heuvel, R. A. Leese and M. A. Shepherd, *Graph labeling and radio channel assignment*, J. Graph Theory, 29 (1998), pp. 263–283.

[19] M. A. Whittlesey, J. P. Georges and D. W. Mauro, *On the* λ*-number of* Q_n *and related graphs*, SIAM J. Discrete Math., 8 (1995), pp. 499–506.

SIAM J. Discrete Math.
Vol. 19, No. 1, pp. 224–244

REVERSALS AND TRANSPOSITIONS OVER FINITE ALPHABETS*

A. J. RADCLIFFE[†], A. D. SCOTT[‡], and E. L. WILMER[§]

Abstract. Extending results of Christie and Irving, we examine the action of reversals and transpositions on finite strings over an alphabet of size k. We show that determining reversal, transposition, or signed reversal distance between two strings over a finite alphabet is NP-hard, while for "dense" instances we give a polynomial-time approximation scheme. We also give a number of extremal results, as well as investigating the distance between random strings and the problem of sorting a string over a finite alphabet.

Key words. strings, sorting, genome comparison, reversals, transpositions, NP-complete problems, MAX-SNP hardness, approximation algorithms

AMS subject classifications. 68R05, 68R15, 68Q17

DOI. 10.1137/S0895480103433550

Introduction. As a result of interest in both modelling large-scale genome changes and fundamental questions on the combinatorics of sequences, rearrangement operations, including transpositions, reversals, and signed reversals, have recently been the focus of intense combinatorial, algorithmic, and complexity-theoretic study. These superficially similar sequence operations turn out to have significantly different properties. Most previous work has concentrated on applying sequence operations to permutations. However, the analysis of operations on strings over finite alphabets was raised by Pevzner and Waterman [26] and investigated by Christie and Irving [9]. The study of sequence operations on strings may also be of some practical interest; for a recent example, see, for instance, Skaletsky et al. [27] on the roles played by palindromes and repetitive segments in the Y-chromosome.

The operations under consideration all act on strings $\alpha = a_1 \cdots a_n$ of length $|\alpha| = n$. The *reversal* R_{ij}, where $i < j$, reverses the substring $a_i \cdots a_j$, so that

$$R_{ij}(a_1 \cdots a_n) = a_1 \cdots a_{i-1} a_j a_{j-1} \cdots a_i a_{j+1} \cdots a_n.$$

The *transposition* T_{ijk}, where $i < j < k$, exchanges the substrings $a_i \cdots a_j$ and $a_{j+1} \cdots a_k$, so

$$T_{ijk}(a_1 \cdots a_n) = a_1 \cdots a_{i-1} a_{j+1} \cdots a_k a_i \cdots a_j a_{k+1} \cdots a_n.$$

The *pancake flip* or *prefix reversal* P_i reverses the substring $a_1 \cdots a_i$, so

$$P_i(a_1 \cdots a_n) = a_i a_{i-1} \cdots a_1 a_{i+1} \cdots a_n.$$

Signed reversals work on strings where each character has an orientation: we use $\overline{a}$ to denote the opposite orientation of a, and note that $\overline{\overline{a}} = a$. The *signed reversal* S_{ij} is

*Received by the editors August 20, 2003; accepted for publication (in revised form) June 9, 2004; published electronically August 3, 2005.

http://www.siam.org/journals/sidma/19-1/43355.html

[†]Department of Mathematics, University of Nebraska-Lincoln, Lincoln, NE 68588-0323 (jradclif@math.unl.edu). This author's research was supported by NSF grant DMS 9401351.

[‡]Department of Mathematics, University College London, Gower Street, London WC1E 6BT, UK (scott@math.ucl.ac.uk).

[§]Department of Mathematics, Oberlin College, Oberlin, OH 44074 (elizabeth.wilmer@oberlin.edu). This author's research was supported by an NSF-AWM Mentoring Travel Grant.

the same as R_{ij}, except that the reversed elements change orientation:

$$S_{ij}(a_1, \ldots, a_n) = a_1 \cdots a_{i-1}\overline{a_j}\,\overline{a_{j-1}} \cdots \overline{a_i}a_{j+1} \cdots a_n.$$

As the collections of reversals, transpositions, and pancake flips each generate the symmetric group S_n and are closed under taking inverses, they therefore induce metrics d_{rev}, d_{tr}, d_{pf} on S_n, where $d_X(\alpha, \beta)$ is the minimum length of a sequence of operations of type X transforming α to β. Signed reversals generate the larger hyperoctahedral group of signed permutations and define a metric $d_{\overline{\mathrm{rev}}}$. All these metrics can be defined for strings over finite alphabets, provided we restrict ourselves to *compatible* pairs, namely pairs of strings that have the same number of occurrences of each symbol.

Extremal investigation of sequence operations has concentrated on the diameter of the symmetric (or, for signed reversals, hyperoctahedral) group. Bafna and Pevzner [2] showed that the reversal diameter of S_n is $n-1$, while Meidanis, Walter, and Dias [25] showed that the signed reversal diameter of the group of signed permutations is at most $n+1$. For transpositions, Bafna and Pevzner [3] showed that the diameter lies between $n/2+1$ and $3n/4$. Eriksson et al. [10] improved the upper bound to $\lfloor (2n-2)/3 \rfloor$ for $n \geq 9$. For pancake flips, Gates and Papadimitriou [14] showed that the diameter lies between $17n/16$ and $(5n+5)/3$; Heydari and Sudborough [19] improved the lower bound to $15n/14$. Christie and Irving [9] investigated these problems for the set of binary strings: they showed that the maximum reversal and transposition distances between two compatible binary strings of length n is $\lfloor n/2 \rfloor$, and noted that there does not appear to be an easy generalization of these results to strings over alphabets of size $k > 2$. In section 1, we prove such a generalization for reversal distance between strings over alphabets of size k; furthermore, we determine the diameter of every equivalence class of strings (under the relation of compatibility).

In section 2, we consider the distance between random strings. Two randomly chosen permutations are typically reversal distance $\Theta(n)$ apart [2]. We show that strings from a k-letter alphabet with fixed fractions of letters of each type are typically at reversal distance $\Theta(n/\log n)$. Our arguments extend to any other class of string operations with a bounded number of cutpoints at each step and a linear bound on diameter in the permutation case, such as transpositions or pancake flips.

The complexity of calculating the distance between two permutations depends on the type of operations used. Caprara [7] showed that determining reversal distance is NP-hard, while Berman and Karpinski [6] (see also [22]) showed that the problem is MAX-SNP hard. The signed reversal distance, by contrast, can be found in polynomial time: algorithms were given by Hannenhalli and Pevzner [17], Berman and Hannenhalli [4], and Kaplan, Shamir, and Tarjan [21]. The complexity of finding transposition distance between permutations remains open. For binary strings, Christie and Irving [9] showed that reversal distance remains NP-hard for binary strings, but left open the difficulty of finding transposition distance. In section 3, we show that signed reversal distance and transposition distance are both NP-hard for binary strings (and hence for strings over any finite alphabet). This is the first hardness result for transposition distance; together with the difficulty of signed reversal distance, it suggests that these problems may be harder over finite alphabets.

In section 4, we turn to the problem of approximating the distance between pairs of strings. Karpinski [22] showed that it is NP-hard to approximate the reversal distance between two permutations to within any factor less than 1237/1236. A number of authors have given approximation algorithms: Kececioglu and Sankoff [23] gave a 2-approximation algorithm, Bafna and Pevzner [2] gave a 1.75-approximation algorithm, Christie [8] gave a 1.5-approximation algorithm, and recently, Berman, Hannenhalli, and Karpinski [5] gave a 1.375-approximation algorithm. Bafna and Pevzner [3] have also given a 1.5-approximation algorithm for transposition distance. For strings over a finite alphabet, Pevzner and Waterman [26, Problem 4], raised the problem of finding an approximation algorithm for determining reversal distance. It follows from Karpinski's results [22] and the results of section 2 that it is NP-hard to approximate reversal distance between strings to within any factor better than 1237/1236. However, we show that for *dense* instances (pairs of strings at distance $\Omega(n)$) there is a polynomial-time approximation scheme. Similar results hold for approximating signed reversal, prefix reversal, or transposition distance between two strings, and we conjecture that analogous results should hold for calculating the distance between permutations.

For permutations, a sorting algorithm suffices to determine the distance between an arbitrary pair of strings—just relabel the entries of both so that one string is sorted. For strings over a finite alphabet this equivalence fails, and sorting is strictly a special case of finding distance. In section 5, we show that the number of reversals required to sort a ternary string can be found in polynomial time. We also give some elementary bounds on reversal sorting over an arbitrary finite alphabet; these restrict any instance of sorting to a finite range of values. We conjecture that, for fixed k, these problems can be solved for k-ary alphabets in polynomial time.

Notation. Our alphabet will generally be the set $[k] = \{1, 2, \ldots, k\}$. We consider strings over this alphabet, elements $\alpha \in [k]^*$. We write $|\alpha|$ for the length of a string. We write $\mathcal{L}(a_1, \ldots, a_k)$ for the set of strings of length n with exactly a_i occurrences of i for each i. (Note that the set of permutations can be thought of as $\mathcal{L}(1, 1, \ldots, 1)$.)

1. Reversal diameter for finite alphabets.

Our approach to finding the reversal diameter of $\mathcal{L}(a_1, \ldots, a_k)$ is straightforward: we present an algorithm that turns one element of $\mathcal{L}(a_1, \ldots, a_k)$ into any other in at most the desired number of reversals, and we also present a pair of elements of $\mathcal{L}(a_1, \ldots, a_k)$ that are provably at least the desired number of reversals apart. In order to prove the lower bound, we introduce an invariant of strings, *tilt*, which is linear in a certain sense and which cannot be changed very much by a single reversal. Bafna and Pevzner [2] looked at properties of a difference graph to prove lower bounds on reversal distance between permutations, and Christie and Irving [9] gave algorithmic upper bounds on the reversal distance between pairs of binary strings. However, both the graph we use to compute tilt and the algorithm we give for our upper bound are different from earlier work.

Given a graph G with vertex set $V \subset \mathbb{Z}^+$ and an edge-weight function $w : \binom{V}{2} \to \mathbb{Z}$, define the *tilt* of G to be

$$t(G) = \sum_{i \text{ odd},\ j \text{ even}} w(\{i,j\})\varepsilon_{ij}, \qquad \text{where } \varepsilon_{ij} = \begin{cases} 1, & i < j, \\ -1, & i > j. \end{cases}$$

Tilt is linear in the following sense: when G and H are weighted graphs on the same vertex set G, let $G + H$ denote the weighted graph on V whose edge weight function is the sum of those of G and H. Then

$$t(G + H) = t(G) + t(H).$$

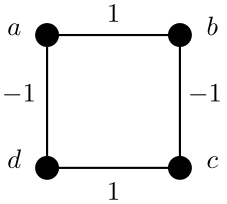


FIG. 1. *A typical alternating square.*

An *alternating square* C on vertices $abcd$ is the weighted graph obtained from the closed walk $abcda$ by giving the edges ab and cd weight 1 and the edges bc and da weight -1 (when the edges are not distinct, the weights are summed; however, we exclude loops).

LEMMA 1. *If C is an alternating square, then $|t(C)| \leq 2$.*

Proof. Label C as shown in Figure 1. We argue by contradiction, first assuming that $t(C) \leq -3$; the other case can be argued symmetrically.

When $t(C) \leq -3$, at least 3 edges must contribute -1 to $t(C)$. Without loss of generality let them be ab, bc, and cd. When a is even,

- $b > a$ and b is odd, hence
- $c > b$ and c is even, hence
- $d > c$ and d is odd.

Thus the edge da contributes $+1$ and $t(C) = -2$. A similar argument applies when a is odd. ▯

Why is this relevant? Given a string $\alpha = \alpha_1 \cdots \alpha_n \in [k]^n$, define the associated weighted graph $G(\alpha)$ to have vertex set $[k]$ and edge weights

$$w(\{i,j\}) = |\{l : \{\alpha_l, \alpha_{l+1}\} = \{i,j\}\}| .$$

That is, $w(e)$ counts the number of times the edge e is used, in either direction, by the walk α. We ignore loops, however. When the reversal R_{ij} is applied to α, the transitions $\alpha_{i-1}\alpha_i$ and $\alpha_j\alpha_{j+1}$ are replaced by $\alpha_{i-1}\alpha_j$ and $\alpha_i\alpha_{j+1}$, while all other transitions remain unchanged. Thus

$$G(R_{ij}(\alpha)) = G(\alpha) + C,$$

where C is an alternating square on vertices $\alpha_i\alpha_{j+1}\alpha_j\alpha_{i-1}$. It follows that if β is obtained from α by a sequence of d reversals, then

$$G(\beta) = G(\alpha) + C_1 + C_2 + \cdots + C_d,$$

where $C_1, \ldots, C_d$ are alternating squares. Linearity of tilt and Lemma 1 now yield the following.

LEMMA 2. *When $\alpha, \beta \in \mathcal{L}(a_1, \ldots, a_k)$,*

$$d_{\text{rev}}(\alpha, \beta) \geq \frac{1}{2} \left| t\left(G(\alpha) - G(\beta)\right) \right| .$$

We use this to determine the diameter of $\mathcal{L}(a_1, \ldots, a_k)$.

THEOREM 3. *The reversal diameter of $\mathcal{L}(a_1, \ldots, a_k) \subseteq [k]^n$ is $n - \max_i a_i$.*

Proof. Without loss of generality, we assume $a_1 \geq a_2 \geq \cdots \geq a_k$. For the upper bound, we give a procedure for successively modifying two strings $\alpha = \alpha_1 \cdots \alpha_n$ and $\beta = \beta_1 \cdots \beta_k$ in $\mathcal{L}(a_1, \ldots, a_k)$ until both are the same. Because reversals are

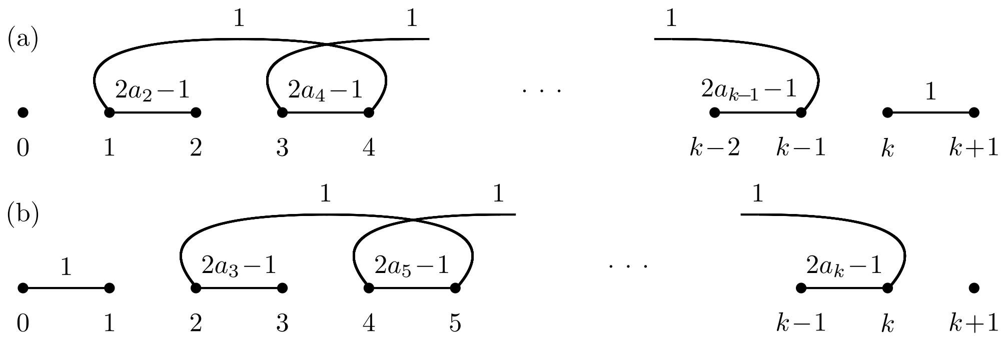


FIG. 2. *Multiplicities of edges joining vertices of opposite parity in* (a) $G(\alpha')$ *and* (b) $G(\beta')$ *(when k is odd).*

involutions on $[k]^n$, we can produce a sequence of the same total length carrying α to β.

Let $\alpha^{(0)} = \alpha$ and $\beta^{(0)} = \beta$. Given $\alpha^{(i)}$ and $\beta^{(i)}$, let j be the smallest index such that $\alpha^{(i)}_j \neq \beta^{(i)}_j$.

- If $\beta^{(i)}_j \neq 1$, pick a $j' > j$ such that $\alpha^{(i)}_{j'} = \beta^{(i)}_j$. Let $\alpha^{(i+1)} = R_{j,j'}(\alpha^{(i)})$ and let $\beta^{(i+1)} = \beta^{(i)}$.
- If $\beta^{(i)}_j = 1$ (and thus $\alpha^{(i)}_j \neq 1$), pick a $j' > j$ such that $\beta^{(i)}_{j'} = \alpha^{(i)}_j$. Let $\beta^{(i+1)} = R_{j,j'}(\beta^{(i)})$ and let $\alpha^{(i+1)} = \alpha^{(i)}$.

For each i, let $\gamma^{(i)}$ be the initial segment on which $\alpha^{(i)}$ and $\beta^{(i)}$ agree. Note that $|\gamma^{(i)}|$ is strictly increasing in i, and that furthermore $|\{j : \gamma^{(i)}_j \neq 1\}|$ is strictly increasing in i. This process must therefore stop after at most

$$|\{j : \alpha^{(0)}_j \neq 1\}| = n - a_1$$

steps.

For the lower bound, we give two strings $\alpha, \beta \in \mathcal{L}(a_1, \ldots, a_k)$ at distance at least $n - a_1$. We actually apply Lemma 2 to $\alpha' = 0\alpha(k+1)$ and $\beta' = 0\beta(k+1)$; since any sequence of reversals taking α to β also takes α' to β', $d_{\text{rev}}(\alpha, \beta) \geq d_{\text{rev}}(\alpha', \beta')$.

When k is odd, take

$$\alpha' = 0(21)^{a_2} 1^{a_1 - a_2} \cdots (k-1\ k-2)^{a_{k-1}} (k-2)^{a_{k-2} - a_{k-1}} k^{a_k} (k+1)$$

and

$$\beta' = 0 1^{a_1} (32)^{a_3} 2^{a_2 - a_3} \cdots (k\ k-1)^{a_k} (k-1)^{a_{k-1} - a_k} (k+1).$$

The edges joining vertices of opposite parity determine tilt; these are shown with their multiplicities in Figure 2 (recall that both ij and ji substrings contribute to $w(\{i,j\})$). As all relevant edges in $G(\alpha')$ increase from odd to even and all relevant edges in $G(\beta')$ increase from even to odd,

$$\begin{aligned} t\big(G(\alpha') - G(\beta')\big) &= (2a_2 - 1) + 1 + (2a_4 - 1) + 1 + \cdots + (2a_{k-1} - 1) + 1 \\ &\qquad + 1 + (2a_3 - 1) + 1 + (2a_5 - 1) + \cdots + 1 + (2a_k - 1) \\ &= 2\sum_{i=2}^{k} a_i = 2(n - a_1). \end{aligned}$$

Lemma 2 now gives the desired lower bound on $d_{\text{rev}}\,(\alpha, \beta)$.

When k is even, the strings

$$0(21)^{a_2}1^{a_1-a_2}\cdots(k\ k-1)^{a_k}(k-1)^{a_{k-1}-a_k}(k+1),$$

$$01^{a_1}(32)^{a_3}2^{a_2-a_3}\cdots(k-1\ k-2)^{a_{k-1}}(k-2)^{a_{k-2}-a_{k-1}}k^{a_k}(k+1)$$

yield the same bound. □

Remark. As we noted earlier, permutations are simply $\mathcal{L}(1, 1, \ldots, 1)$. In this case, our argument gives a new proof that the reversal diameter of S_n is $n - 1$. If the symbols appearing in α and β are relabeled so that β is taken to $123\cdots n$, then α is taken to $315274\cdots$ (with ending depending on the parity of n). Bafna and Pevzner [2] showed that this permutation and its inverse are the only permutations at distance $n - 1$ from the identity.

We do not have an analogue of Theorem 3 for transpositions.

Problem 1. *What is the transposition diameter of $\mathcal{L}(a_1, \ldots, a_k)$?*

There are similar problems for signed reversals, prefix reversals, etc.

2. Distance between random strings. The diameter of S_n is $\Theta(n)$ for each of the families of transformations we are considering: reversals [24], transpositions [3, 10], pancake flips [14]. The distance between randomly chosen permutations can quickly be seen to be $\Theta(n)$ with high probability, since each family has a bounded number of cuts per transformation and two random permutations have only about Poisson(2) adjacencies in common.

For strings taken from a finite alphabet with a positive fraction of the string devoted to each letter, we have shown above that the diameter under reversals is linear. Cutpoint arguments clearly imply the same for the other families of operations. In this section we show that the distance between random strings σ_1, σ_2 in the same component of $[k]^n$ is typically much smaller: only $\Theta(n/\log n)$. First, both σ_1 and σ_2 are partitioned into substrings of length approximately $c \log n$. With high probability, most of the resulting pieces appear about the same number of times in σ_1 and σ_2. For each family of operations considered, these substrings can be arranged and any remaining letters aligned in $O(n/\log n)$ operations. Furthermore, with high probability σ_1 and σ_2 have no common substrings of length $C \log n$, where $C > c$, and thus at least $n/(C \log n)$ cuts must be made.

Most of this section examines the anatomy of pairs of random strings. Our conclusions about distances between typical pairs can be drawn for any collection of operations with boundedly many cutpoints per operation.

Let $\mathbf{p} = (p_1, \ldots, p_k)$ be a rational probability vector satisfying $p_1 \geq p_2 \geq \cdots \geq p_k$ and let $h_i = -\log p_i$. For $\alpha = \alpha_1 \cdots \alpha_m \in [k]^*$, let $h(\alpha) = \sum_{i=1}^{m} h_{\alpha_i}$ be the entropy of α. We also set $H = H(\mathbf{p}) = \sum_{i \in [k]} p_i h_i$, the entropy of $\mathbf{p}$.

Given a $c > 0$, define the *c-threshold set*, $\mathcal{A}_c$, to consist of all words $\alpha = \alpha_1 \dots \alpha_m \in [k]^*$ such that $h(\alpha) \geq c \log n$, but $h(\alpha_1 \dots \alpha_{m'}) < c \log n$ for $m' < m$. (Much of the notation in this section conceals dependence on n.)

The following are immediate from definitions.

$$\text{For } \alpha \in \mathcal{A}_c, \quad c \log n \leq h(\alpha) < c \log n + h_k. \tag{1}$$

$$\text{For } \alpha \in \mathcal{A}_c, \quad \frac{c \log n}{h_k} \leq |\alpha| \leq \left\lceil \frac{c \log n}{h_1} \right\rceil. \tag{2}$$

$$n^c \leq |\mathcal{A}_c| < \frac{n^c}{p_k}. \tag{3}$$

We also note that

$$\sum_{\alpha \in \mathcal{A}_c} e^{-h(\alpha)} = 1. \tag{4}$$

This simply states that for the random process $\alpha_1 \alpha_2 \dots$, in which the α_i are chosen independently according to $\mathbf{p}$, the stopping time

$$T = \min\{m : h(\alpha_1 \alpha_2 \dots \alpha_m) \geq c \log n\}$$

has $\Pr(T < \infty) = 1$.

The next lemma will be useful as we decompose strings into short pieces.

LEMMA 4. *Let* $L_c = \sum_{\alpha \in \mathcal{A}_c} |\alpha| e^{-h(\alpha)}$ *be the expected value of the length of a random* $\alpha \in \mathcal{A}_c$ *determined by successive i.i.d. choices of letters according to* $\mathbf{p}$. *Then*

$$L_c = \frac{c \log n}{H} \left(1 + o\left(1\right)\right).$$

Proof. Consider characters $\alpha_1, \alpha_2, \dots$ chosen independently from $[k]$ according to $\mathbf{p}$. For each i, $E[h_{\alpha_i}] = H$; let $\sigma^2 = \mathrm{Var}[h_{\alpha_i}]$.

Let $\alpha^- = \alpha_1 \cdots \alpha_{m-}$ and $\alpha^+ = \alpha_1 \cdots \alpha_{m^+}$ be the initial strings of lengths $m^- = \left\lceil \frac{c \log n}{H} - (\log n)^{2/3} \right\rceil$ and $m^+ = \left\lceil \frac{c \log n}{H} + (\log n)^{2/3} \right\rceil$, respectively, and let $\alpha = \alpha_1 \dots \alpha_{|\alpha|} \in \mathcal{A}_c$. The definition of $\mathcal{A}_c$ and Chebyshev's inequality now give

$$\begin{aligned} \Pr\left[|\alpha| \leq m^-\right] &= \Pr\left[h(\alpha^-) \geq c \log n\right] \\ &= \Pr\left[h(X^-) - Hm^- > H(\log n)^{2/3} + O(1)\right] \\ &\leq \frac{\sigma^2 m^-}{(H(\log n)^{2/3} + O(1))^2} = O((\log n)^{-1/3}). \end{aligned} \tag{5}$$

Similarly,

$$\begin{aligned} \Pr\left[|\alpha| > m^+\right] &= \Pr\left[h(X^+) < c \log n\right] \\ &= \Pr\left[h(X^+) - Hm^+ < -H(\log n)^{2/3} + O(1)\right] \\ &\leq \frac{\sigma^2 m^+}{(H(\log n)^{2/3} + O(1))^2} = O((\log n)^{-1/3}). \end{aligned} \tag{6}$$

Since (2), (5), and (6) show that $|\alpha|$ is within $O((\log n)^{2/3})$ of $c \log n/H$ with probability at least $1 - O((\log n)^{-1/3})$ and is always within a bounded factor of $c \log n$, we have

$$\begin{aligned} L_n &= \left(1 - O((\log n)^{-1/3})\right)\left(\frac{c\log n}{H} + O((\log n)^{2/3})\right) + O\left(\frac{\log n}{(\log n)^{1/3}}\right) \\ &= \frac{c\log n}{H}\left(1 + o(1)\right). \qquad \square \end{aligned}$$

In what follows, we always let $n \to \infty$ in such a way that $p_1 n, \ldots, p_k n$ are all integral.

THEOREM 5. *Fix $\epsilon > 0$. When σ_1 and σ_2 are chosen independently and uniformly from $\mathcal{L}(p_1 n, \ldots, p_k n)$, then, with probability approaching 1 as $n \to \infty$, σ_1 and σ_2 can be broken into identical collections of at most $(1+\epsilon)(\frac{Hn}{\log n})$ substrings.*

Proof. We first consider two strings, ρ_1 and ρ_2, each consisting of n letters chosen independently from $[k]$ according to $\mathbf{p}$. We show that ρ_1 and ρ_2 can be broken into words of approximately equal probability in such a way that both strings have approximately equal numbers of each type of word. We then modify ρ_1 and ρ_2 slightly to obtain σ_1 and σ_2, each uniformly distributed in $\mathcal{L}(a_1, \ldots, a_k)$; as these modifications do not affect very many pieces, all discrepancies can be broken into singletons.

Fix a $\delta \in (0,1)$ such that $\frac{1}{1-\delta} < 1 + \epsilon$, and call $\alpha \in [k]^*$ *substantial* when $\alpha \in \mathcal{A}_{1-\delta}$. Each ρ_j, $j = 1, 2$, can be broken uniquely into disjoint substantial words, starting from the left and proceeding down the string. We call those words $\beta_1^j, \beta_2^j, \ldots$. (We can regard each ρ_j as the initial segment of length n from an infinite string chosen according to $\mathbf{p}$, and we extract the β_i^j from this infinite string. Thus β_i^j is defined for all i.) Let

$$N = \frac{n}{L_{1-\delta}} - \sqrt{n}.$$

We claim that, with high probability, each ρ_j contains at least N substantial words, while the first N substantial words of each ρ_j cover at least $N - \sqrt{n}(\log n)^2$ characters. Chebyshev's inequality, together with (2), gives

$$\begin{aligned} \Pr\left[|\beta_1^j| + \cdots + |\beta_N^j| > n\right] &\le \Pr\left[|\beta_1^j| + \cdots + |\beta_N^j| - NL_{1-\delta} > L_{1-\delta}\sqrt{n}\right] \\ &\le \frac{N(O((\log n)^2))}{nL_{1-\delta}^2} = O((\log n)^{-1}) = o(1) \end{aligned}$$

and

$$\begin{aligned} &\Pr\left[|\beta_1^j| + \cdots + |\beta_N^j| < n - \sqrt{n}(\log n)^2\right] \\ &\qquad \le \Pr\left[|\beta_1^j| + \cdots + |\beta_N^j| - NL_{1-\delta} < -\sqrt{n}(\log n)^2(1 + o(1))\right] \\ &\qquad \le \frac{O(N(\log n)^2)}{n(\log n)^4} = O((\log n)^{-3}) = o(1). \end{aligned}$$

For each $\alpha \in \mathcal{A}_{1-\delta}$, let $N_{\alpha,j}$ be the number of pieces of type α among the first N substantial words of ρ_j. We use the following Chernoff-type inequality (see Janson, Łuczak, and Rucinski [20, p. 26]: if $X \sim \text{binomial}(n,p)$, then for $t > 0$,

$$\Pr[|X - EX| > t] \le 2\exp\left(-\frac{t^2}{np + \frac{t}{3}}\right).$$

Since $N_{\alpha,j}$ is binomial$(N, e^{-h(\alpha)})$,

$$\begin{aligned}
\Pr[|N_{\alpha,j} - E(N_{\alpha,j})| \geq n^{\delta/2}\log n]& \\
&\leq 2\exp\left(-\frac{n^{\delta}(\log n)^2}{2\left(\frac{Hn(1+o(1))}{(1-\delta)\log n}\left(\frac{1}{n^{1-\delta}}\right)+\frac{n^{\delta/2}\log n}{3}\right)}\right) \\
&= 2\exp\left(-\Omega\left((\log n)^3\right)\right) = o\left(\frac{1}{n^{1-\delta}}\right).
\end{aligned}$$

Now sum over $\alpha \in \mathcal{A}_{1-\delta}$ and $j = 1, 2$. Equation (3) implies

$$\Pr\left[\max_{\alpha,i} |N_{\alpha,i} - E(N_{\alpha,i})| < n^{\delta/2}\log n\right] \to 1 \text{ as } n \to \infty.$$

Thus, we can with high probability match up all except

$$n^{\delta/2}\log n|\mathcal{A}_{1-\delta}| = O\left(n^{1-\delta/2}\log n\right)$$

of the first N substantial words in ρ_1 with (distinct) counterparts among the first N substantial words of ρ_2.

We now modify ρ_1 and ρ_2 to obtain uniformly distributed elements σ_1 and σ_2 of $\mathcal{L}(a_1, \ldots, a_k)$. For $i \in [k]$, let $N_{i,j}$ denote the number of i's in ρ_j. Since $N_{i,j}$ is a binomial(n, p_i) random variable, Chebyshev gives

$$\Pr\left[|N_{i,j} - p_i n| > \sqrt{n}\log n\right] \leq \frac{p_i(1-p_i)n}{(\log n)^2 n} = O((\log n)^{-2}),$$

so

$$\Pr\left[\max_{i,j} |N_{i,j} - p_i n| < \sqrt{n}\log n\right] \to 1 \text{ as } n \to \infty.$$

To generate σ_1 and σ_2, we must reallocate some sites containing overrepresented characters to currently underrepresented ones. Take as many sites as necessary uniformly from each overrepresented letter, and fill the entire collection of sites thus selected with an assignment of the appropriate multiset of characters uniformly chosen from the possible assignments. With high probability, we need only change $O(\sqrt{n}\log n)$ characters—and thus will break at most that many of the substantial-word matches we built between ρ_1 and ρ_2.

Now break all unmatched substantial words (including those past position N that were never considered, those among the first N that we tried to match but failed, and those whose matches were broken by character modifications) in both σ_1 and σ_2 into single characters. Let N^* be the resulting number of fragments in each string. With high probability as $n \to \infty$,

$$\begin{aligned}
N^* &= N + O\left(\sqrt{n}(\log n)^2 + n^{1-\delta/2}(\log n)^2 + \sqrt{n}(\log n)^2\right) \\
&= \frac{Hn}{(1-\delta)\log n}(1+o(1)) \leq (1+\epsilon)Hn
\end{aligned}$$

for n sufficiently large. □

Theorem 6. *Fix $\delta > 0$. Choose σ_1 and σ_2 independently and uniformly from $\mathcal{L}(p_1 n, \ldots, p_k n)$. For $i = 1, 2$, let $\mathcal{S}_i$ be the multiset of all substrings of σ_i that belong to $\mathcal{A}_{1+\delta}$. Then*

$$\Pr[|\mathcal{S}_1 \cap \mathcal{S}_2| \geq n^{1-\delta}(\log n)] \to 0 \textit{ as } n \to \infty.$$

Proof. For $\alpha \in [k]^*$, let q_α be the probability that α occurs as an initial substring of a string σ chosen uniformly from $\mathcal{L}(p_1 n, \ldots, p_k n)$. We can generate such a σ by sampling without replacement from a bag containing $p_i n$ copies of i. From this it is easy to see that, for c fixed and $\alpha \in \mathcal{A}_c$,

$$q_\alpha \leq \prod_{i=1}^{|\alpha|} \left(p_{\alpha_i} \left(\frac{n}{n - |\alpha|} \right) \right) = e^{-h(\alpha)}(1 + o(1)) \leq n^{-c}(1 + o(1)). \tag{7}$$

(The $o(1)$ estimate follows from (3).) Thus the probability that α appears as a substring of σ starting at any given position is also at most $n^{-c}(1 + o(1))$.

Since there are only $n - O(\log n)$ possible starting positions in σ_1 for a substring of weight at least $1 + \delta$, we trivially have $|\mathcal{S}_1| \leq n$. Similarly, there are $n - O(\log n)$ possible starting positions in σ_2 for a substring in $\mathcal{A}_{1+\delta}$. Let N be the number of locations i for which the corresponding substring is an element of $\mathcal{S}_1$; clearly $N \geq |\mathcal{S}_1 \cap \mathcal{S}_2|$. By (7), for any given $\mathcal{S}_1$,

$$E[N|\mathcal{S}_1] \leq n \sum_{\alpha \in \mathcal{S}_1} q_\alpha \leq n^{1-\delta}(1 + o(1)),$$

so

$$E[N] \leq n^{1-\delta}(1 + o(1)),$$

and Markov's inequality gives

$$\Pr[N > n^{1-\delta} \log n] = o(1). \qquad \square \tag{8}$$

The following theorem combines the previous results to give bounds on the distance between random strings.

Theorem 7. *Fix $\epsilon > 0$, and choose σ_1 and σ_2 uniformly and independently from $\mathcal{L}(p_1 n, \ldots, p_k n)$. Then each of the following statements holds with probability approaching 1 as $n \to \infty$:*

(i) $\frac{1-\epsilon}{2}(\frac{Hn}{\log n}) \leq d_{\text{rev}}(\sigma_1, \sigma_2) \leq (1 + \epsilon)(\frac{Hn}{\log n})$.

(ii) $\frac{1-\epsilon}{3}(\frac{Hn}{\log n}) \leq d_{\text{tr}}(\sigma_1, \sigma_2) \leq \frac{2(1+\epsilon)}{3}(\frac{Hn}{\log n})$.

(iii) $(1 - \epsilon)(\frac{Hn}{\log n}) \leq d_{\text{pf}}(\sigma_1, \sigma_2) \leq 2(1 + \epsilon)(\frac{Hn}{\log n})$.

Proof. For the upper bounds, we need only apply results on the diameter of S_n under the various operations to the decomposition of σ_1 and σ_2 into at most $N^* = (1 + \epsilon)\frac{Hn}{\log n}$ identical pieces that Theorem 5 provides with high probability. Meidanis, Walter, and Dias [24] show that the signed reversal diameter of S_{N^*} is at most $N^* + 1$, while Eriksson et al. [10] bounded the transposition diameter of S_{N^*} by $\lfloor (2N^* - 2)/3 \rfloor$ for $N^* > 9$, and Gates and Papadimitriou [14] showed that the signed pancake-flipping diameter of S_{N^*} is at most $2N^* + 3$.

The lower bounds are nearly as simple. Fix a $\delta > 0$ such that $1 - \epsilon < \frac{1}{1+\delta}$. We call a word $\alpha \in \mathcal{A}_{1+\delta}$ *unusual*, while a word $\alpha \in \mathcal{A}_3$ is termed *implausible*. Equations (3) and (7) guarantee that

$$\begin{aligned}\Pr[\sigma_1 \text{ and } \sigma_2 \text{ share an implausible substring}] &\le n^2 \left(\frac{n^3}{p_k}\right)\left(\frac{(1+o(1))}{n^3}\right)^2 \\ &= O(n^{-1}).\end{aligned}$$

Equation (2) implies that, with probability $1 - O(1/n)$, all common substrings of σ_1 and σ_2 have length at most $\lceil \frac{3\log n}{h_1} \rceil$. If we (temporarily) define the *weight* of a word $\alpha \in [k]^*$ to be $h(\alpha)/\log n$, then we can compute as follows. By Theorem 6, there are, with high probability, fewer than $n^{1-\delta}\log n$ sites in each string to start common unusual substrings. Equation (8) now implies that with high probability at most $n^{1-\delta}(\log n)\lceil \frac{3\log n}{h_1} \rceil = o(n)$ characters are contained in common substrings of weight greater than $1+\delta$ (and their combined weight is also $o(n)$). Any sequence of operations taking σ_1 to σ_2 must cut the remaining characters into words of weight less than $1+\delta$. The entire string, without the unusual common substrings, has weight $(H - o(1))n/\log n$, and so there must be at least $(1+o(1))\frac{Hn}{(1+\delta)\log n}$ cuts. A single reversal makes at most two cuts, a single transposition makes at most three cuts, and a single pancake flip makes at most one cut, giving the results above. ∎

We conjecture that, in each case, the expected value of $d(\sigma_1, \sigma_2)/(n/\log n)$ tends to a constant; we also leave open the further problem of determining this constant.

3. Complexity.

The complexity of sorting permutations by reversals or transpositions has been extensively studied. Kececioglu and Sankoff [23] conjectured that sorting reversals by permutations (MIN-SBR) is NP-hard, and this was proved by Caprara [7]. Berman and Karpinski [6] showed that sorting by reversals is MAX-SNP hard; Karpinski [22] showed that for any $\epsilon > 0$ it is NP-hard to approximate reversal distance within a factor $1237/1236 - \epsilon$. A number of authors have given approximation algorithms: Kececioglu and Sankoff [23] gave a 2-approximation algorithm, Bafna and Pevzner [2] gave a 1.75-approximation algorithm, Christie [8] gave a 1.5-approximation algorithm, and recently, Berman, Hannenhalli, and Karpinski [5] gave a 1.375-approximation algorithm.

The problem of sorting *signed* permutations by reversals turns out to be polynomial time, as was shown by Hannenhalli and Pevzner [17]. Faster algorithms were found by Berman and Hannenhalli [4] and by Kaplan, Shamir, and Tarjan [21].

The complexity of sorting by transpositions remains unknown, although Bafna and Pevzner [3] have given a 1.5-approximation algorithm.

Christie and Irving [9] considered the complexity of reversal distance and transposition distance for strings over finite alphabets. They showed that reversal distance is NP-hard for binary strings, although a binary string can be sorted in polynomial time. They also showed that binary strings can be sorted by transpositions in polynomial time, but left open the complexity of transposition distance.

We begin this section by giving another proof of Christie and Irving's [9] result that reversal distance is NP-hard for strings over binary alphabets; this of course implies that determining reversal distance is NP-hard for any finite alphabet. Using a similar argument we also show that, surprisingly, signed reversal distance is also NP-hard for signed strings over finite alphabets. We then prove that sorting by transpositions is NP-hard for binary strings.

THEOREM 8. *Reversal distance is NP-hard for binary strings.*

Proof. We give a reduction from sorting permutations by reversals. Given a permutation $\pi = \pi(1) \cdots \pi(n)$, we define the string $\lambda(\pi)$ by

$$\lambda(\pi) = (10^{\pi(1)}1)^{2n} \cdots (10^{\pi(n)}1)^{2n}.$$

We call the substrings $(10^{\pi(i)}1)^{2n}$ the *blocks* of $\lambda(\pi)$. Each block consists of $2n$ *subblocks*, each of the form $10^{\pi(i)}1$. Clearly, $\lambda(\pi)$ can be constructed from π in polynomial time.

Given permutations π_1 and π_2, it is easy to see that

$$d_{\text{rev}}\left(\lambda(\pi_1), \lambda(\pi_2)\right) \leq d_{\text{rev}}\left(\pi_1, \pi_2\right),$$

since a sequence of reversals mapping π_1 to π_2 maps to a sequence of reversals on the corresponding sequence of blocks $(10^{\pi_1(j)}1)^{2n}$ in $\lambda(\pi_1)$ (note that each block is invariant under reversals).

Now let $t = d_{\text{rev}}\left(\lambda(\pi_1), \lambda(\pi_2)\right)$. If $t < d_{\text{rev}}\left(\pi_1, \pi_2\right)$, then consider a sequence of t reversals taking $\lambda(\pi_1)$ to $\lambda(\pi_2)$. Since the reversal diameter of S_n is less than n, we have $t < n$. Now consider a block $(10^{\pi_1(i)}1)^{2n}$. This contains $2n$ subblocks: since the t reversals cut the string in at most $2t < 2n$ places, there must be one subblock I_i that does not get cut. It follows that I_i must get mapped to a segment of the block $(10^{\pi_1(i)}1)^{2n} = (10^{\pi_2(i')}1)^{2n}$, where $i' = \pi_2^{-1}\pi_1(i)$.

Thus the segments $I_1, \ldots, I_n$, which occur in order in $\lambda(\pi_1)$, are rearranged by the sequence of t reversals to occur in $\lambda(\pi_2)$ in the order $I_{\pi_1^{-1}\pi_2(1)}, \ldots, I_{\pi_1^{-1}\pi_2(n)}$. Considering the action of the reversals just on the segments $I_1, \ldots, I_n$ implies that there exists a sequence of t reversals which rearranges id to $\pi_1^{-1}\pi_2$. Since $d_{\text{rev}}\left(id, \pi_1^{-1}\pi_2\right) = d_{\text{rev}}\left(\pi_1, \pi_2\right)$, this is a contradiction.

We therefore have $d_{\text{rev}}\left(\lambda(\pi_1), \lambda(\pi_2)\right) = d_{\text{rev}}\left(\pi_1, \pi_2\right)$, and so we have a reduction from reversal distance for permutations to reversal distance for binary strings. ▯

As noted above, signed permutations can be sorted in polynomial time [17, 4, 21]. By contrast, over finite alphabets, the problem of finding signed reversal distance is NP-hard.

THEOREM 9. *Signed reversal distance is NP-hard for binary strings.*

Proof. As in the previous theorem, we reduce from MIN-SBR. Given a permutation $\pi = \pi(1) \cdots \pi(n)$, we encode π by

$$\lambda(\pi) = (10^{\pi(1)}1\overline{1}\,\overline{0}^{\pi(1)}\overline{1})^{2n} \cdots (10^{\pi(n)}1\overline{1}\,\overline{0}^{\pi(n)}\overline{1})^{2n}.$$

Since each block is invariant under reversal,

$$d_{\overline{\text{rev}}}\left(\lambda(\pi_1), \lambda(\pi_2)\right) \leq d_{\text{rev}}\left(\pi_1, \pi_2\right).$$

Arguing as before, we deduce that

$$d_{\overline{\text{rev}}}\left(\lambda(\pi_1), \lambda(\pi_2)\right) = d_{\text{rev}}\left(\pi_1, \pi_2\right).$$

Thus signed reversal distance is NP-hard. ▯

As we have seen, signed reversal distance is NP-hard for strings with repeated symbols. We show now that the difficulty remains even if we allow only two occurrences of each symbol.

THEOREM 10. *Signed reversal distance is NP-hard for strings in which there is at most one positive and one negative occurrence of each symbol.*

Proof. We prove this by reduction from MIN-SBR. We start by mapping each instance $\pi(1) \cdots \pi(n)$ to the string $S = \pi(1)\overline{\pi(1)} \cdots \pi(n)\overline{\pi(n)}$. The signed distance from this to $T = 1\bar{1}\cdots n\bar{n}$ is clearly at most the reversal distance from $\pi(1)\cdots\pi(n)$ to $1\cdots n$ (note that $i\bar{i}$ is reversal-invariant). On the other hand, any sequence of signed reversals taking S to T yields a corresponding sequence of reversals taking $\pi(1)\cdots\pi(n)$ to $1\cdots n$ by restricting attention to the (unsigned) action of the signed reversals on the elements of S that initially have positive orientation and then ignoring signs. The two distances are therefore equal and so it is NP-hard to determine signed reversal distance. □

We remark that if only $O(\log n/\log\log n)$ repeats in total are allowed, then signed reversal distance is solvable in polynomial time, since we can systematically examine all possible pairings between symbols of the same type; on the other hand, it is NP-hard to determine signed reversal distance with $\Omega(n^\epsilon)$ repeats, since we can encode instances of MIN-SBR of size n^ϵ using the methods of Theorem 10 and then pad out with additional variables occurring once each and in the same order at the end of each string.

THEOREM 11. *Transposition distance is NP-hard for binary strings.*

Proof. We prove the result first for strings over the alphabet $\{0, 1, 2, 3\}$ by reduction from the NP-hard problem 3-PARTITION [13], which we quote here.

INSTANCE: Positive integers n and N and positive integers $a_1, \dots, a_{3n}$, with $N/4 < a_i < N/2$ for every i.

QUESTION: Is there a partition of the a_i into n (multi)sets of size 3, each summing to N?

The problem 3-PARTITION is strongly NP-hard [13]: that is, there is a polynomial $p(n)$ such that it is still NP-hard when all the a_i are at most $p(n)$. Our reduction is polynomially bounded for instances of this type.

Given an instance of 3-PARTITION with weights $a_1, \dots, a_{3n}$ bounded by $p(n)$, consider the strings

$$S = 2^{n+1}30^{a_1}130^{a_2}13\cdots 30^{a_{3n}}13$$

and

$$T = (20^N1^3)^n 23^{3n+1}.$$

Note that if this instance of 3-PARTITION is solvable, then $d_{\text{tr}}(S, T) \le 3n$, since we can successively generate each segment 20^N1^32 of T by moving three intervals $0^{a_i}1$ between an adjacent pair of 2's. On the other hand, suppose that $d_{\text{tr}}(S, T) \le 3n$. Note that, among the adjacencies in S, the sequence must destroy n adjacencies of form 22, $3n$ adjacencies of form 30, $3n$ adjacencies of form 13, and $2n$ adjacencies of form 01, a total of $9n$ adjacencies. It follows that there must be exactly $3n$ transpositions in the sequence, and furthermore that no transposition cuts an adjacency 00. The blocks 0^{a_i} of zeros in S are therefore preserved in T and so constitute a solution to 3-partition.

We have shown that $d_{\text{tr}}(S, T) = 3n$ if and only if our instance of 3-PARTITION has a solution. Since the lengths of S and T are bounded by a polynomial in n, it follows that it is NP-hard to determine transposition distance over alphabets of size 4.

To show that transposition distance is NP-hard for binary strings, we use an encoding similar to that in Theorem 8. Given a string $\epsilon = \epsilon_1\cdots\epsilon_n$ with $\epsilon \in \{0, 1, 2, 3\}$, we encode it as

$$\lambda(\epsilon) = (10^{\epsilon_1+1}1)^{3n}\cdots(10^{\epsilon_n+1}1)^{3n}.$$

For strings ϵ and ϵ' of length n, since $d_{\mathrm{tr}}(\epsilon, \epsilon') \leq n-1$ and each transposition cuts the string in three places, arguing as before, we find that

$$d_{\mathrm{tr}}(\lambda(\epsilon), \lambda(\epsilon')) = d_{\mathrm{tr}}(\epsilon, \epsilon').$$

Composing these two reductions, both of which are polynomially computable for instances whose components are of polynomially bounded magnitude, sends instances of 3-PARTITION to instances of transposition distance for binary strings. We conclude that transposition distance for binary strings is NP-hard. $\square$

Let us note that the reductions from MIN-SBR employed in Theorems 8 and 9 are distance-preserving. Since MIN-SBR is NP-hard to approximate to within any factor better than 1237/1236 [22], it follows that reversal distance and signed reversal distance for binary strings are also NP-hard to approximate to within better than 1237/1236. We conjecture that, for some $\epsilon > 0$, it is NP-hard to approximate transposition distance for binary strings to within a factor better than $1+\epsilon$.

4. An approximation algorithm for dense instances.

For many NP-hard approximation problems, it is also NP-hard to find an approximate solution that is correct to within a small multiplicative factor. However, it is sometimes easier to handle *dense* cases of these problems. For instance, although there is an algorithm that approximates Max Cut to within a factor 1.138 [15], it is NP-hard to approximate to within a factor better than 17/16 [18]. On the other hand, for dense instances of Max Cut (instances G with $\Omega(|G|^2)$ edges or minimum degree $\Omega(|G|)$), it is possible to find a polynomial-time approximation scheme [11, 12]. Similar results exist for a number of other NP-hard problems (see, for instance, [1, 22]).

Our aim in this section is to describe a polynomial-time approximation scheme for dense instances of reversal distance for strings over a finite alphabet. This requires us to define a notion of "density" for instances of reversal distance: for $c > 0$, we say that an instance (σ, τ) with $|\sigma| = |\tau| = n$ is *c-dense* if $d_{\mathrm{rev}}(\sigma, \tau) \geq cn$. We show below that, for any fixed k and any $\epsilon > 0$, there is a linear time algorithm that approximates reversal distance between k-ary strings of length n to within an additive error ϵn. It follows that, for fixed k, and any $c > 0$ and $\epsilon > 0$, there is a linear time algorithm that approximates reversal distance of c-dense instances to within a factor $1+\epsilon$.

We first prove a simple lemma concerning the effect of deletions on reversal distance.

LEMMA 12. *Suppose that σ and τ are compatible strings and that σ' and τ' are compatible strings obtained by deleting m elements from each string. Then*

$$|d_{\mathrm{rev}}(\sigma', \tau') - d_{\mathrm{rev}}(\sigma, \tau)| \leq 2m.$$

Proof. To see that

$$d_{\mathrm{rev}}(\sigma, \tau) \leq d_{\mathrm{rev}}(\sigma', \tau') + 2m,$$

consider an optimal sequence of reversals taking σ' to τ'. These same reversals can be applied to σ and τ, always placing cuts that occur in sites where letters have been deleted to the left of those letters. The two-reversal sequence shown below suffices to move an individual letter to a new location without changing the rest of the string:

$$A|xB|C \to A|\overline{B}|xC \to ABxC.$$

Thus we can correct the reinserted letters with at most $2m$ additional reversals. A similar argument shows that

$$d_{\mathrm{rev}}(\sigma', \tau') \leq d_{\mathrm{rev}}(\sigma, \tau) + 2m. \qquad \square$$

The existence of a polynomial-time approximation scheme follows immediately from the following theorem.

Theorem 13. *For each fixed k and every $\epsilon > 0$ there is a linear time algorithm that approximates reversal distance between k-ary strings of length n to within an additive error of ϵn and outputs a sequence of reversals achieving this bound.*

Proof. The main idea of the proof is to break up the problem instance into "good" subinstances of a finite number of types, each of which can be sorted optimally. The subinstances can then be recombined at small cost.

Given $\epsilon < 1/10$, let $K = \lceil 100/\epsilon \rceil$ and let $\alpha_1, \dots, \alpha_L$ be an enumeration of the $L = k^K$ k-ary strings of length K. Let $f : [L] \to \mathbb{R}^k$ count the number of each digit present in the strings α_i: thus $(f(i))_j$ is equal to the number of times j occurs in α_i. We now build a mapping $\tilde{f} : \mathbb{R}^L \to \mathbb{R}^k$ by setting $\tilde{f}(a_1, \dots, a_L) = \sum_{i=1}^{L} a_i f(i)$. Thus if we break a string X into segments of length K, obtaining a_i segments of type α_i for each i, then $\tilde{f}(a_1, \dots, a_L)$ counts the number of occurrences of each character in the original string X.

For $d \geq 1$, we write U_d for the unit simplex in $\mathbb{R}^d$ given by the convex hull of the origin and the d standard unit basis vectors. Let $D = \{x_1, \dots, x_M\} \subseteq U_k$ satisfy the following two conditions:

(i) D is an $\epsilon/4$-net in U_k. (That is, every point in U_k is distance less than $\epsilon/4$ from some $x_i \in D$.)

(ii) All the coordinates of each $x_i \in D$ are rational.

For $i = 1, \dots, M$, let $X_i = f^{-1}(x_i)$ be the affine subspace in $\mathbb{R}^L$ that maps to x_i and let E_i be an $\epsilon/4K$-net in $X_i \cap U_L$, where we choose E_i so that each point has all coordinates rational.

By deleting at most $(\epsilon/4)n$ elements from the string σ, we obtain a string σ' of length $n' < n$ such that, for some i, σ' has $(x_i)_j n'$ occurrences of the digit j for $1 \leq j \leq k$. Delete the same collection of characters from τ to obtain τ'. Then σ' and τ' are compatible strings; furthermore, we may assume that both σ' and τ' have length a multiple of K.

Now break each of σ' and τ' into segments of length K. Deleting at most $\epsilon n'/4K$ segments from each of σ' and τ', we may assume that we have strings σ'' and τ'', each broken into $n'' \geq (1-\epsilon)n'/K$ segments of length K such that σ'' has $n'' a_j$ copies of α_j and τ'' has $n'' b_j$ copies of α_j for each j, where a and b both belong to E_i. By the definition of E_i, σ'' and τ'' are compatible. Furthermore, by Lemma 12, $|d_{\text{rev}}(\sigma'', \tau'') - d_{\text{rev}}(\sigma, \tau)| < \epsilon n/2$.

It is therefore sufficient to solve the following problem to within an additive constant ϵn.

- INPUT: Two elements a and b of E_i and two compatible strings σ and τ such that
  - $|\sigma| = |\tau| = n$, where $K|n$;
  - when broken into segments of length K, σ falls into $a_i n/K$ copies of S_i for each i;
  - when broken into words of length K, τ falls into $b_i n/K$ copies of S_i for each i.
- OUPUT: An optimal sequence of reversals taking σ to τ.

This breaks up into a constant number of separate problems, one for each choice of $a, b \in E_i$. We show that each of these problems has a good approximation algorithm.

Fix i and let $a, b \in E_i$. Let n_0 be an integer such that all entries of $n_0 a$ and $n_0 b$ are integers. Note that we can rearrange the n/K segments of σ and τ into any order with at most $4n/K \leq \epsilon n/25$ reversals. We can therefore assume that the segments of σ and τ are in any order we choose, with cost at most $\epsilon n/25$.

For $j \geq 1$, let A_j be the collection of strings of length $n_0 jK$ constructed from $n_0 j a_l$ copies of α_l for each l, in any order; similarly, let B_j be the set of strings with $n_0 j b_l$ copies of α_l for each l, again in any order. Let

$$d_j = \min\{d_{\text{rev}}\,(\alpha, \beta) : \alpha \in A_j, b \in B_j\}.$$

Clearly, for $j, j' \geq 1$,

$$d_{j+j'} \leq d_j + d_{j'}, \tag{9}$$

since strings from A_j and $A_{j'}$ can be concatenated to form strings in $A_{j+j'}$. Thus d_j is subadditive and therefore d_j/j tends to a limit r_∞. Let j_0 be large enough so that $|d_j/j - r_\infty| \leq \epsilon/4$ for $j \geq j_0$, and let $\alpha^{(0)} \in A_{j_0}$, $\beta^{(0)} \in B_{j_0}$ be strings with $d_{\text{rev}}\left(\alpha^{(0)}, \beta^{(0)}\right) = d_{j_0} \leq j_0 r_\infty + \epsilon j_0/4$.

Now for $m \geq 1$, given strings $\alpha \in A_m$ and $\beta \in B_m$, we can rearrange α in blocks of size K to give α' with $\lfloor m/j_0 \rfloor$ copies of $\alpha^{(0)}$ and at most a constant number of additional segments; similarly, we can rearrange β to get β' with $\lfloor m/j_0 \rfloor$ copies of $\beta^{(0)}$ and at most a constant number of additional segments. Each of these rearrangements takes at most $n/K \leq \epsilon n/100$ reversals. Furthermore,

$$d_{\text{rev}}\,(\alpha', \beta') \leq \left\lfloor \frac{m}{j_0} \right\rfloor d_{\text{rev}}\left(\alpha^{(0)}, \beta^{(0)}\right) + O(1) = \frac{m d_{j_0}}{j_0} + O(1),$$

and we can write down an explicit sequence of reversals taking α to β in this time. Finally, note that

$$d_{\text{rev}}\,(\alpha, \beta) \geq m r_\infty \geq \frac{m d_{j_0}}{j_0} - \frac{\epsilon m}{4} \geq \frac{m d_{j_0}}{j_0} - \frac{\epsilon n}{K},$$

so our approximation is correct to within ϵn. □

The following corollary is an immediate consequence of the theorem.

Corollary 14. *For every fixed k and $c > 0$, there is a polynomial-time approximation scheme for c-dense instances of reversal distance for k-ary strings.*

Note that a similar argument gives a polynomial-time approximation scheme for dense instances of transposition distance over finite alphabets. The same approach also works for prefix reversals, except that (9) is replaced by

$$d_{j+j'} \leq d_j + d_{j'} + O(1),$$

as we work on the concatenation AB by first working on A, then reversing the entire string and working on B. This implies that d_j/j approaches some limit r_∞ (for instance, as an easy corollary of a result of Hammersley [16]), and the rest of the argument goes through with minor modification.

Given these results for strings over finite alphabets, it is natural to ask what happens for permutations. Recall that, for permutations, it suffices to consider MIN-SBR, the problem of sorting, for which each instance is a single permutation. We say that a permutation σ of length n is *c-dense* if there are at least cn integers i with $1 \leq i < n$ such that $|\sigma(i) - \sigma(i+1)| > 1$. It follows that c-dense strings necessarily require $\Omega(n)$ reversals to sort. We conjecture that dense permutations exhibit the same behavior as dense pairs of strings.

Conjecture 1. *For every $c > 0$, there is a polynomial-time approximation scheme for c-dense instances of sorting permutations by reversals.*

We also conjecture that similar statements hold for sorting permutations by prefix reversals and sorting permutations by transpositions.

5. Sorting strings over finite alphabets. Any algorithm for determining the number of reversals necessary to sort a permutation suffices to determine the reversal distance between an arbitrary pair of permutations: the distance between π_1 and π_2 is the same as that between $\pi_1\pi_2^{-1}$ and *id*. For strings from a finite alphabet, there is no such correspondence. There is, however, some hope that this special case of the distance problem might be easier than the full one. Indeed, Christie and Irving [9] show that the number of reversals required to sort a binary string σ is one less than the number of 0-blocks in the string 0σ, which can of course easily be determined in polynomial time. (An *i-block* is a maximal nonempty substring consisting entirely of copies of the character i.) We give below a similarly simple criterion for determining the number of reversals required to sort a ternary string, along with some elementary bounds over an arbitrary finite alphabet.

For the remainder of this section, we generally prepend a 0 and append a $k-1$ to all strings in $\{0, 1, \dots, k-1\}^*$. These added characters are not allowed to move, but are included when we count blocks. We also generally replace each i-block with a single copy of the character i in example strings. The string resulting from applying both operations to σ is called the *standard form* of σ. Note that the number of reversals required to sort the standard form of σ is identical to the number required to sort σ.

We call a reversal *optimal* if it reduces the number of blocks by 2. Every optimal reversal is of the form $\dots a|b \dots a|b \dots$. Conversely, whenever the string contains a *repeated transition*—that is, some substring containing two distinct characters occurs more than twice in the string—an optimal reversal is possible.

It will be convenient to categorize transitions between pairs of consecutive characters by the *unordered* pair of characters involved. There are then three types of transitions: 01/10, 02/20, and 12/21. For example, 01212101202 contains three 01/10 transitions, two 02/20 transitions, and five 12/21 transitions.

Define a 02*-block* to be a maximal substring consisting only of 0's and 2's and containing at least one of each. Call a 02-block *odd* if it contains an odd number of blocks of 0's and 2's; otherwise, call it *even.*

Example. The standard form of 0122000002222112111020 is 012021210202, which has two 02-blocks; the first is odd and the second is even.

Lemma 15. *Let σ be a ternary string whose standard form has b blocks. If b is odd (even), then σ has an even (odd) number of* 02/20 *transitions, while the numbers of* 01/10 *and* 12/21 *transitions are both odd (even).*

Proof. Consider the standard form of σ to be a walk on the vertices $\{0, 1, 2\}$. Since the walk begins at 0 and ends at 2, the vertices 0 and 2 have odd degree, while vertex 1 has even degree. Thus the numbers of 01/10 and 12/21 transitions are of the same parity, which is opposite to that of both the number of 02/20 transitions and the total number of transitions. □

We say that a string σ satisfies the *matching odd block condition* if σ has at least one 02-block, all 02-blocks of S are odd, and all 02-blocks of S have the same initial character.

Theorem 16. *Let σ be a ternary string containing all* 3 *possible characters whose standard form has b blocks. Then the minimal number of reversals required to sort σ*

is $\lceil \frac{b-3}{2} \rceil$*, unless* σ *satisfies the matching odd block condition, in which case sorting* σ *requires* $\lceil \frac{b-3}{2} \rceil + 1$ *reversals.*

Proof. We note first that each reversal can reduce the number of blocks by at most 2. Thus $\lceil \frac{b-3}{2} \rceil$ is necessarily a lower bound for the number of reversals required to sort σ.

When b is even, Lemma 15 ensures that σ does not satisfy the matching odd block condition. It is not difficult to sort directly in this case. As long as there are repeated transitions, perform optimal reversals—which do not change the parity of the block number. Since there are only 6 possible transitions over the 3-letter alphabet, the resulting string σ' has at most 6 transitions. Let b' be the number of blocks in σ'. As $3 \leq b' \leq 7$ and b' is even, it is true that $b' = 4$ or $b' = 6$. The only possible structures for b' are shown below, along with an optimal reversal sorting of each:

$$0|10|2 \to 0012,$$

$$0|21|2 \to 0122,$$

$$0|10|212 \to 001|21|2 \to 001122,$$

$$0|1210|2 \to 001|21|2 \to 001122,$$

$$0|210|12 \to 001|21|2 \to 001122.$$

Hence we can sort σ in $\frac{b-b'}{2} + \lceil \frac{b'-3}{2} \rceil = \lceil \frac{b-3}{2} \rceil$ reversals, matching the elementary lower bound.

Now assume b is odd, but σ does not satisfy the matching odd block condition. We proceed in order through each of the following steps.

- Use optimal reversals internal to single 02-blocks to reduce any 02-blocks with 4 or more blocks to either 2 or 3 sub-blocks:

  $$\cdots 0|20|2 \cdots \to \cdots 0022 \cdots .$$

  These reversals do not change the parities of the 02-blocks. After this stage, all remaining 02 blocks have one of the following forms: 02, 20, 202, or 020.
- Because σ did not originally satisfy the matching odd block condition and we have not changed the parities or types of any 02-blocks, either there are odd 02-blocks of both types or there are even 02-blocks. In the case that there are odd 02-blocks with different initial characters, we reduce them via an optimal reversal to two even 02-blocks:

  $$\cdots 02|0 \cdots 2|02 \cdots \to \cdots 022 \cdots 002 \cdots .$$

  After this step, we may assume that there are even 02-blocks present.
- An even 02-block and an odd 02-block can be reduced with an optimal reversal to a single even 02-block:

  $$\cdots 2|02 \cdots 2|0 \cdots \to \cdots 22 \cdots 200 \cdots$$

  or

  $$\cdots 20|2 \cdots 0|2 \cdots \to \cdots 200 \cdots 22 \cdots .$$

  We repeat this step until all remaining odd 02-blocks are eliminated.

- As long as there are at least three even 02-blocks, some pair must match and an optimal reversal can eliminate both.
- We now have at most two even 02-blocks remaining. By Lemma 15, the number of even 02-blocks must be even. If the last two match, they can be eliminated in a single optimal reversal. If not, we are in one of the cases

  $$0\cdots 02\cdots 20\cdots 2 \quad\text{or}\quad 0\cdots 20\cdots 02\cdots 2.$$

  In the first case, the final 2-block must be preceded by a 1-block (since we have eliminated all other 02 transitions); in the second, the initial 0-block must be followed by a 1-block. Once we fill in those and the 1-blocks that must border the even 02-blocks, it becomes clear that for either case there is an optimal reversal that reverses one of the even 02-blocks, after which both 02-blocks can be eliminated:

  $$0\cdots 02\cdots 1|201\cdots 1|2 \qquad\text{or}\qquad 0|1\cdots 120|1\cdots 02\cdots 2.$$

Once all 02/20-transitions have been eliminated, there are only 01/10- and 12/21-transitions. By Lemma 15, there is an odd number of each. Apply optimal reversals until the number of each is reduced to one. Since the string starts with 0 and ends with 2, the remaining transitions must be 01 and 12, and the string is sorted.

Finally, assume that σ satisfies the matching odd block condition. An easy case analysis shows that any string resulting from the application of an optimal reversal to σ also satisfies the matching odd block condition, and so cannot be completely sorted. Thus at least one nonoptimal reversal must be used when sorting σ, and the number of reversals required to do so is strictly greater than $\frac{b-3}{2}$.

In order to sort σ, first apply optimal reversals until a string σ' containing no repeated transitions is obtained and let b' be the number of blocks of σ'. We know that the string σ' still satisfies the matching odd block condition and thus is not sorted, so $4 \le b' \le 7$. We also know the number of 02/20 transitions is even, so by Lemma 15 the numbers of 01/10 and 12/21 transitions are both odd, and $b' = 7$ is impossible. Thus, $b' = 5$. The only possible types of 5-block strings are shown below, each with an optimal reversal sorting:

$$0|120|2 \to 00|21|2 \to 00122,$$

$$0|20|12 \to 00|21|2 \to 00122.$$

We have sorted σ in $\left\lceil \frac{b-5}{2} \right\rceil + 2 = \left\lceil \frac{b-3}{2} \right\rceil + 1$ reversals. □

Remark. In the b odd case, naive choices of optimal reversals can get one into trouble. For instance, the string 021021202 does not satisfy the matching odd block condition and so can be sorted in 3 reversals. Applying the optimal reversal 0[210]21202 results in 001221202, which does satisfy the matching odd block condition—and thus itself requires 3 reversals.

What about strings from larger alphabets? We can combine earlier observations to determine the number of reversals required to sort a string from a k-ary alphabet up to a finite error (whose magnitude depends on k).

THEOREM 17. *Let σ be a k-ary string whose standard form contains all k letters and has b blocks, and let t be the number of reversals required to sort σ. Then*

$$\left\lceil \frac{b-k}{2} \right\rceil \le t \le \left\lceil \frac{b-k}{2} \right\rceil + \frac{k(k-1)}{4} + 1.$$

Proof. The lower bound is clear; each reversal can reduce the number of blocks by at most 2. For the upper bound, we first use optimal reversals to reduce to a string σ' with r blocks and no repeated transitions; r must have the same parity as b. By Theorem 3, σ' can be sorted in at most $r - \lceil \frac{r}{k} \rceil$ steps. We've shown that

$$\begin{aligned} t &\leq \frac{b-r}{2} + \left\lceil \frac{r-k}{2} \right\rceil + \left(r - \left\lceil \frac{r}{k} \right\rceil - \left\lceil \frac{r-k}{2} \right\rceil \right) \\ &\leq \left\lceil \frac{b-k}{2} \right\rceil + \left(\frac{1}{2} - \frac{1}{k} \right) r + \frac{k}{2}. \end{aligned}$$

Now substitute $r \leq \binom{k}{2} + 1$ to obtain the desired inequality. □

CONJECTURE 2. *For each fixed k, the number of reversals required to sort k-ary strings can be determined in time polynomial in string length.*

It also seems plausible that there are polynomial-time algorithms for determining the number of operations required to sort k-ary strings using transpositions or pancake flips.

Acknowledgments. This work was partially completed during visits by the last two authors to the University of Nebraska-Lincoln and the Isaac Newton Institute for Mathematical Sciences. We are grateful to both institutions for their hospitality.

REFERENCES

[1] S. Arora, D. Karger, and M. Karpinski, *Polynomial time approximation schemes for dense instances of NP-hard problems*, J. Comput. System Sci., 58 (1999), pp. 193–210.

[2] V. Bafna and P. A. Pevzner, *Genome rearrangements and sorting by reversals*, SIAM J. Comput., 25 (1996), pp. 272–289.

[3] V. Bafna and P. A. Pevzner, *Sorting by transpositions*, SIAM J. Discrete Math., 11 (1998), pp. 224–240.

[4] P. Berman and S. Hannenhalli, *Fast sorting by reversal*, in Combinatorial Pattern Matching (Laguna Beach, CA, 1996), Springer, Berlin, 1996, pp. 168–185.

[5] P. Berman, S. Hannenhalli, and M. Karpinski, 1.375-*Approximation Algorithm for Sorting by Reversals*, Technical report, DIMACS TR:2001-41, Piscataway, NJ, 2001.

[6] P. Berman and M. Karpinski, *On some tighter inapproximability results (extended abstract)*, in Automata, Languages and Programming (Prague, 1999), Springer, Berlin, 1999, pp. 200–209.

[7] A. Caprara, *Sorting by reversals is difficult*, in Proceedings of the First International Conference on Computational Molecular Biology, ACM Press, New York, 1997, pp. 75–83.

[8] D. A. Christie, *A 3/2-approximation algorithm for sorting by reversals*, in Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (San Francisco, CA, 1998), ACM, New York, 1998, pp. 244–252.

[9] D. A. Christie and R. W. Irving, *Sorting strings by reversals and by transpositions*, SIAM J. Discrete Math., 14 (2001), pp. 193–206.

[10] H. Eriksson, K. Eriksson, J. Karlander, L. Svensson, and J. Wästlund, *Sorting a bridge hand*, Discrete Math., 241 (2001), pp. 289–300.

[11] W. Fernandez de la Vega, *Max-cut has a randomized approximation scheme in dense graphs*, Random Structures Algorithms, 8 (1996), pp. 187–198.

[12] W. Fernandez de la Vega and M. Karpinski, *Polynomial time approximation of dense weighted instances of MAX-CUT*, Random Structures Algorithms, 16 (2000), pp. 314–332.

[13] M. R. Garey and D. S. Johnson, *Computers and Intractibility*, W. H. Freeman, San Francisco, 1979.

[14] W. H. Gates and C. H. Papadimitriou, *Bounds for sorting by prefix reversal*, Discrete Math., 27 (1979), pp. 47–57.

[15] M. X. Goemans and D. P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. ACM., 42 (1995), pp. 1115–1145.

[16] J. M. Hammersley, *Generalization of the fundamental theorem on sub-additive functions*, Proc. Cambridge Philos. Soc., 58 (1962), pp. 235–238.

[17] S. Hannenhalli and P. A. Pevzner, *Transforming cabbage into turnip: Polynomial algorithm for sorting signed permutations by reversals*, J. ACM, 46 (1999), pp. 1–27.

[18] J. Hastad, *Some optimal inapproximability results*, in STOC '97 (El Paso, TX), ACM, New York, 1999, pp. 1–10.

[19] M. H. Heydari and I. H. Sudborough, *On the diameter of the pancake network*, J. Algorithms, 25 (1997), pp. 67–94.

[20] S. Janson, T. Łuczak, and A. Rucinski, *Random Graphs*, Wiley-Interscience, New York, 2000.

[21] H. Kaplan, R. Shamir, and R. E. Tarjan, *A faster and simpler algorithm for sorting signed permutations by reversals*, SIAM J. Comput., 29 (2000), pp. 880–892.

[22] M. Karpinski, *Polynomial time approximation schemes for some dense instances of NP-hard optimization problems*, Algorithmica, 30 (2001), pp. 386–397.

[23] J. Kececioglu and D. Sankoff, *Exact and approximation algorithms for sorting by reversals, with application to genome rearrangement*, Algorithmica, 13 (1995), pp. 180–210.

[24] J. Meidanis, M. E. M. T. Walter, and Z. Dias, *A Lower Bound on the Reversal and Transposition Diameter*, Technical report IC-00-16, Institute of Computing, University of Campinas, Campinas, Brazil, 2000.

[25] J. Meidanis, M. E. M. T. Walter, and Z. Dias, *Reversal Distance of Signed Circular Chromosomes*, Technical report IC-00-23, Institute of Computing, University of Campinas, Campinas, Brazil, 2000.

[26] P. A. Pevzner and M. S. Waterman, *Open combinatorial problems in computational molecular biology*, in Third Israel Symposium on the Theory of Computing and Systems (Tel Aviv, 1995), IEEE Comput. Soc. Press, Los Alamitos, CA, 1995, pp. 158–173.

[27] H. Skaletsky et al., *The male-specific region of the human Y chromosome is a mosaic of discrete sequence classes*, Nature, 423 (2003), pp. 825–837.

SIAM J. Discrete Math.
Vol. 19, No. 1, pp. 245–257

DIRECTED NETWORK DESIGN WITH ORIENTATION CONSTRAINTS[*]

SANJEEV KHANNA[†], JOSEPH (SEFFI) NAOR[‡], AND F. BRUCE SHEPHERD[§]

Abstract. We study directed network design problems with *orientation constraints*. An orientation constraint on a pair of nodes u and v states that a feasible solution may include at most one of the arcs (u, v) and (v, u). Such constraints arise naturally in many network design problems, since link or edge resources such as fiber can be used to support traffic in one of two possible directions only. Our first result is that the directed network design problem with orientation constraints can be solved in polynomial time in the case where the requirement function f is positively intersecting supermodular. (The case where there are no orientation constraints follows from the work of Frank [*Acta Sci. Math. (Szeged)*, 41 (1979), pp. 63–76].) The second main result of the paper is a 4-approximation algorithm for the minimum cost strongly edge-connected subgraph problem with orientation constraints. Our algorithm shows that the problem of enforcing orientation constraints can be reduced to the minimum cost 2-edge-connected subgraph problem on undirected graphs. Finally, we study the problem for general crossing supermodular functions and show the following bicriteria approximation result. Let k denote the maximum requirement of any set under the given requirement function f. We give a $2k$-approximation algorithm to construct a solution that satisfies a slightly weaker requirement function, namely, $f'(S) = \max\{f(S) - 1, 0\}$.

Key words. connectivity, orientation, submodular flow, approximation algorithms

AMS subject classifications. 68Q25, 68W25, 90C59

DOI. 10.1137/S0895480100380112

1. Introduction.

We study directed network design problems in mixed networks with *orientation constraints*. We are given a directed graph $D = (V, A)$ and an undirected graph $G = (V, E)$. Let $M = (V, E \cup A)$ denote the *mixed graph* obtained by taking their union. An *orientation* of an undirected edge is obtained by replacing it by one of two possible directed arcs parallel to it. We consider problems which amalgamate two previously studied models for network design. We wish, at minimum cost, to fulfill a given connectivity requirement in a network by (a) selecting a subgraph and (b) orienting its undirected edges. We start by first considering a concrete special case of such orientation and subgraph selection problems.

Let $r \in V$ be a special node, and consider the following two problems, the first concerning D, the second concerning M. Call a digraph *k-edge-connected from r* if it contains k arc-disjoint paths from r to each other node.

(A) Given a cost function on A, find a minimum cost subdigraph (if there is one) which is k-edge-connected from r.

[*] Received by the editors October 26, 2000; accepted for publication (in revised form) December 19, 2003; published electronically August 3, 2005. A preliminary version appeared in the *Proceedings of the* 11*th Annual ACM–SIAM Symposium on Discrete Algorithms*, 2000, pp. 663–671.

http://www.siam.org/journals/sidma/19-1/38011.html

[†] Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104 (sanjeev@cis.upenn.edu). The research of this author was supported in part by an Alfred P. Sloan Research Fellowship.

[‡] Computer Science Department, Technion, Haifa 32000, Israel (naor@cs.technion.ac.il). Most of this work was done while the author was at Bell Laboratories, Lucent Technologies, 700 Mountain Avenue, Murray Hill, NJ 07974. Research supported in part by US-Israel BSF grant 2002276 and by EU contract IST-1999-14084 (APPOL II).

[§] Bell Laboratories, Lucent Technologies, 700 Mountain Avenue, Murray Hill, NJ 07974 (bshep@research.bell-labs.com).

(B) Find an orientation (if there is one) of the undirected edges of M which results in a minimum cost digraph which is k-edge-connected from r.

The polyhedron for problem (A) is described in [6] while that for (B) is derived from the integrality of submodular flow polyhedra (see, e.g., [9, 24]). One aim of the present paper is to describe the polyhedron for the following common generalization.

(AB) Find a mixed subgraph M' of M and orient the undirected edges of M' so that the resulting digraph is k-edge-connected from r, and so that the cost of the directed edges plus the cost of the orienting the undirected edges of M' is minimized.

We study the problem (AB) in the context of the following more general framework.

Input: We are given a directed graph $D = (V, A)$ with a cost function $c : A \rightarrow \mathcal{Z}$, a requirement function f defined over all subsets of V, and a collection $\mathcal{E}$ of disjoint *constrained* arc pairs, each of which induces a *digon* (i.e., two arcs directed in opposite directions). Let $A_p \subseteq A$ denote the set of constrained arcs, and for any arc $a \in A_p$, let a^- denote the arc that a is paired with. We are also given nonnegative lower and upper bound vectors l, u for each pair in $\mathcal{E}$ and each arc a.

Goal: Find an optimal solution to the integer program below; here $\delta^+(S)$ denotes the set of arcs leaving $S \subseteq V$:

$$\text{(I)} \qquad \min \quad \sum_{a \in A} c_a x_a,$$

$$\text{(1)} \qquad \sum_{a \in \delta^+(S)} x_a \geq f(S) \qquad \text{for each } S \subset V,$$

$$\text{(2)} \qquad l_{a,a^-} \leq x_a + x_{a^-} \leq u_{a,a^-} \qquad \text{for each } \{a, a^-\} \in \mathcal{E},$$

$$\text{(3)} \qquad x_a \in \{l_a, l_a + 1, \ldots, u_a\} \quad \text{for each } a \in A.$$

We refer to constraints (1) as the *cut* constraints, constraints (2) as the *orientation* constraints, and constraints (3) as the *integrality* constraints. The term orientation here “arises” from the consideration of constrained pairs with $l_{a,a^-} = u_{a,a^-} = 1$. In this case, the choice of the variables x_a, x_{a^-} amounts to determining the orientation of an associated undirected edge.

The above framework, without the orientation constraints, already captures a large number of fundamental combinatorial optimization problems. Some representative examples include minimum cost branchings, minimum cost k-strongly edge-connected subgraphs, and the directed Steiner network problem. Many of these problems are NP-hard, and thus research is often focused on the design of approximation algorithms for these problems.

In the present paper, we restrict attention to *crossing supermodular* requirement functions f. That is, for every $X, Y \subseteq V$ such that $X \cap Y \neq \emptyset$ and $X \cup Y \neq V$ we have that

$$f(X) + f(Y) \leq f(X \cap Y) + f(X \cup Y).$$

Directed network design problems with a crossing supermodular requirement function remain NP-hard. An example of a crossing supermodular function is the function $f(S) = k$ for all nonempty subsets $S \subset V$; the associated problem is known as the minimum cost k-strongly edge-connected subgraph problem. For this case, a simple 2-approximation algorithm is obtained by solving two minimum cost k-disjoint arborescence problems (one into and one out from) at an arbitrary node v [5].

Frank [6] showed that in the special case where the requirement function is also *intersecting supermodular*, i.e., the above inequality holds whenever X and Y intersect, the network design problem can be solved optimally in polynomial time. Melkonian and Tardos [22] have recently shown that his result can be used to obtain a 2-approximation algorithm for any requirement function which is crossing supermodular.

In undirected graphs, so-called weakly supermodular functions model a broad class of network design problems, including, for instance, the generalized Steiner tree problem. Following a line of work [1, 12, 13, 25], Jain [15] devised an ingenious 2-approximation algorithm for weakly supermodular functions. He proved that every basic feasible solution to the linear programming relaxation of the problem contains a variable of value at least a half. Jain's algorithm finds and rounds such a *large component* iteratively until a final integral solution is obtained.

Network design problems in undirected graphs are often much better understood than their directed counterparts. In particular, techniques for network design problems on undirected graphs, e.g., the widely used primal-dual approach [1, 12], do not seem to be easily amenable to directed network design problems. In the work of Melkonian and Tardos [22] a directed analogue of Jain's result is given which requires significant further insight into the combinatorial structure of basic solutions. They show that every basic solution to a linear programming relaxation of the design problem contains a variable of value at least a quarter, whenever the requirement function f is crossing supermodular. We note that the linear programming relaxation for (I) is polynomially solvable by the ellipsoid method [14].

1.1. Our problems. We study directed network design problems with orientation constraints (as specified in (I)). We can view these as two-phase problems: finding a subgraph of an undirected graph, or a mixed graph, and then orienting its edges so as to satisfy the cut constraints. The cost function associated with the orientation may in general be *asymmetric*; i.e., the cost of orienting an edge $e = uv$ from v to u is different from orienting it from u to v. (An edge can only be oriented in one direction.) The cost of an orientation is defined to be the sum of the costs of the orientations of the edges.

Thus, our network design problems combine constraints of two types, subgraph constraints and orientation constraints. While each type of constraint has been well studied separately, much less is known for design problems that combine these two types of constraints simultaneously.

Perhaps the most basic orientation problem with asymmetric costs that involves both subgraph constraints and orientation constraints is finding, among all subgraphs of G that admit a strong orientation, one that has a strong orientation of minimum cost. This problem generalizes two well-known NP-hard problems. If the orientation cost function is symmetric, then the problem reduces to finding a minimum cost 2-edge-connected subgraph of G. On the other hand, if there are no orientation constraints, then the problem reduces to finding a minimum cost strongly edge-connected subgraph of a directed graph. We note that for both problems, 2-approximation algorithms are known.

One case we study for which a complete solution is given is that where the requirement function is positively intersecting supermodular. This is a generalization of the problem (AB) defined at the outset of the paper. These results appear in section 3.

Another important case of special interest in our study is the design of strongly edge-connected directed graphs with orientation constraints. In this case the requirement function f satisfies $f(X) = 1$ for every proper nonempty subset X, and $f(V) = f(\emptyset) = 0$.

An interesting special case of the asymmetric orientation problem is when the constraints (2) in (I) become $x_a + x_{a^-} = 1$ for each arc $a \in A$. Equivalently, we are given an undirected graph and we need to find a minimum cost orientation that satisfies the requirement function f. A good characterization for the special case of strong edge-connectivity requirements follows from the classical min-max theorem of Lucchesi and Younger [21]. A *dijoin* is a set of arcs in a digraph which intersects every directed cut $\delta^+(S)$, i.e., S such that $\delta^-(S) = \emptyset$. For an integer cost vector c, the Lucchesi–Younger theorem states that the minimum c-cost of a dijoin is equal to the maximum packing of directed cuts where each arc a is allowed in at most c_a cuts in this packing. This theorem also implies the existence of a polynomial time algorithm to find the minimum dijoin via the ellipsoid method; see [14] (see [7] for a combinatorial algorithm). We can find a minimum cost strongly edge-connected orientation as follows. First, orient each constrained pair in the cheaper of its two directions to obtain a digraph D'. We must now reverse some of these orientations in order to make D strongly edge-connected. If A' is the subset of arcs of D' which are flipped, then evidently A' must be a dijoin. So we could do no better than to choose a cheapest such dijoin; but does reversing the arcs of a dijoin result in a strongly edge-connected digraph? In general no, but Lovász [20, Exercise 6.11] and, independently, Younger [26] proved that any minimal dijoin does indeed have this property. The orientation problem for general crossing supermodular requirement functions can also be solved in polynomial time via reductions to submodular flows [3]; see [9, 24]. The reader is referred to [19, 23] for related work on supermodularity problems.

Orientation constraints arise in many network design problems, since link/edge resources, such as fiber, are commonly unidirectional (i.e., they support traffic in only one of the two possible directions at a given time). Asymmetric costs may arise in many network routing problems. For instance, consider the setting where traffic demand is being incrementally introduced in an existing network. Load balancing constraints may favor forcing traffic in opposite directions between a given pair of switches. Hence when routing new demands, costs on the directed links may increase proportionately to the amount of existing traffic. Asymmetric costs may also arise in network planning due to assorted line termination equipment; these are the costs associated with terminating the two ends of a link.

1.2. Our results. Our first result is that if the requirement function f is positively intersecting supermodular, then the extreme points of the relaxation to (I) are integral. This naturally suggests the use of the ellipsoid method together with an oracle for f to design a polynomial time algorithm for the directed network design problem with orientation constraints. One caveat is that the linear programming relaxation need not be solvable in polytime for every *positively* intersecting supermodular function, although this is the case for most common applications. This is because usually such functions arise from intersecting supermodular functions by truncating them to be nonnegative. Recently, however, it was shown that this is not true for all such functions f [18].

We show that any basic solution for the relaxation of (I) in this case has integral components. This generalizes the work of Frank [6], who proved the same result for the variant with no orientation constraints. In fact, we establish this result for the more general formulation given in (I).

Our second result is that the minimum cost strongly edge-connected subgraph problem with orientation constraints has a 4-approximation algorithm. We give a combinatorial approximation algorithm based on the idea that the problem of enforcing orientation constraints can be reduced to the minimum cost 2-edge-connected subgraph problem. We start with any feasible solution to the minimum cost strongly edge-connected subgraph problem and use the above reduction to modify the solution so as to satisfy the violated orientation constraints.

Finally, we study our problem for general crossing supermodular functions and show the following bicriteria approximation result. Let k denote the maximum requirement of any set under the given requirement function f. We give a $2k$-approximation algorithm to construct a solution that satisfies a slightly weaker requirement function, namely, $f'(S) = \max\{f(S) - 1, 0\}$.

2. Preliminaries. We denote a directed graph by $D = (V, A)$. For $S \subseteq V$, denote by $\delta^+(S)$ (respectively, $\delta^-(S)$) the set of arcs with tail in S (respectively, $V - S$) and head in $V - S$ (respectively, S).

A pair of subsets X, Y, of a ground set V, is *intersecting* if $X \cap Y, X - Y, Y - X \neq \emptyset$. An intersecting pair X, Y of sets is *crossing* if $X \cup Y \neq V$ and X, Y are noncomparable. A family $\mathcal{F}$ of nonempty subsets of V is *intersecting* if we have $X \cap Y, X \cup Y \in \mathcal{F}$ for each intersecting pair $X, Y \in \mathcal{F}$. A family is *crossing* if $X \cap Y, X \cup Y \in \mathcal{F}$ for each crossing pair $X, Y \in \mathcal{F}$. A function $f : 2^V \rightarrow \mathcal{Z}_+$ is *positively crossing* (respectively, *positively intersecting supermodular*) on a crossing (respectively, intersecting) family $\mathcal{F}$ if

1. $f(V) = f(\emptyset) = 0$,
2. for each crossing (respectively, intersecting) pair $X, Y \in \mathcal{F}$ such that $f(X)$, $f(Y) > 0$, $f(X) + f(Y) \leq f(X \cap Y) + f(X \cup Y)$.

We emphasize that we only require the inequality to hold for X, Y in the support of f.[1]

For any ordered pair (u, v) of nodes of V, we define an operator Ψ_{uv} as follows. Given any $f : 2^V \rightarrow \mathcal{Q}_+$, $\Psi_{uv}(f)$ is a new function such that $\Psi_{uv}(f)(S) = f(S) - 1$ if $u \in S, v \notin S$, and $f(S) > 0$. Otherwise $\Psi_{uv}(f)(S) = f(S)$. The following is proved in [6] and follows from the submodularity of the in-degree function of a digraph.

LEMMA 2.1. *If f is positively crossing (respectively, positively intersecting) supermodular on $\mathcal{F}$, then $\Psi_{uv}(f)$ is also positively crossing (respectively, positively intersecting) supermodular.*

Note that this result does not hold for crossing (intersecting) supermodular functions (i.e., without the positive requirement). When f is clear from the context, we denote by $\mathcal{F}(uv)$ the family obtained from $\mathcal{F}$ by removing all sets $S \neq V, \emptyset$ for which $\Psi_{uv}(f)(S) = 0$. One easily checks that if f is positively crossing (intersecting) supermodular on $\mathcal{F}$, then it is also positively crossing (intersecting) supermodular on $\mathcal{F}(uv)$.

A family $\mathcal{S} = \{S_i\}_{i=1}^m$ of proper, nonempty subsets of a finite ground set V is *cross-free* if no pair of sets in $\mathcal{S}$ cross. The family is *laminar* if for each pair S_i, S_j of distinct sets in $\mathcal{S}$, we have $S_i \subseteq S_j, S_j \subseteq S_i$, or $S_i \cap S_j = \emptyset$.

3. Intersecting supermodularity. In this section we study the polyhedron obtained by relaxing the integrality constraints (3) in (I), i.e., for each $a \in A$, we

[1] This property has also been referred to as *weakly* crossing (intersecting) supermodularity in [6], whereas supermodularity had referred to functions which satisfy condition (2) for *all* crossing (intersecting) pairs X, Y in $\mathcal{F}$.

now require only $l_a \leq x_a \leq u_a$. In the following, $\mathcal{F}$ denotes an intersecting family of subsets of V, and f is a positively intersecting supermodular function on $\mathcal{F}$.

Let D be a digraph and $\mathcal{E}$ be a disjoint collection of arc pairs, a and a^-, each of which forms a *digon*, i.e., directed circuit of length two. A quadruple $(D, f, \mathcal{F}, \mathcal{E})$ will be called simply an *f-network* (or positively intersecting supermodular network according to f). A *capacitated* f-network is a sextuple $(D, f, \mathcal{F}, \mathcal{E}, l, u)$ where $(D, f, \mathcal{F}, \mathcal{E})$ is an f-network and $l, u : A \cup \mathcal{E} \to \mathcal{Z}_+$ are assignments of capacities to the arcs and digons of $\mathcal{E}$.

For any such capacitated f-network and cost vector c, the *f-edge-connectivity problem* is to find an optimal solution to (I). Denote by $P(D, f, \mathcal{F}, \mathcal{E}, l, u)$ the polyhedron defined by the relaxation of (I), that is replacing (3) by $l_a \leq x_a \leq u_a$ for each arc a. We assume that the polyhedron in consideration is nonempty; in particular, note also that it is pointed and hence has vertices. Each extreme point is thus defined by a system of m linearly independent tight inequalities. We are interested primarily in integer solutions to such capacitated f-connectivity problem and so our goal is to show the following theorem.

THEOREM 3.1. *For any capacitated positively intersecting f-network $(D, f, \mathcal{F}, \mathcal{E}, l, u)$, the extreme points of $P(D, f, \mathcal{F}, \mathcal{E}, l, u)$ are integral.*

The special case where there are no orientation constraints follows from Frank [6]. We note that it is easy to construct examples such that there is an unbounded gap between the cost of optimal solutions to the two versions of the problem with and without orientation constraints.

This theorem first appeared in [16], where a proof based on a primal analysis of the extreme points of $P(D, f, \mathcal{F}, \mathcal{E}, l, u)$ was given. Subsequently, it was communicated to us (independently by J. Cheriyan and A. Frank) that the result can be proved by showing that the system of inequalities associated with $P(D, f, \mathcal{F}, \mathcal{E}, l, u)$ is *totally dual integral* (TDI). That is, for every integer vector c, the dual linear program has an integer optimum. Classical results (due to Hoffman, and Edmonds and Giles) then immediately imply the theorem. Whereas the original primal proof itself may be of some use in other settings, the TDI approach for the present result is so simple and standard we only present this argument here. Moreover, in the meantime, a strengthening of the original result has appeared in [11]. In particular, they extend the result to a hypergraph setting and to orientation constraints over larger sets of possible orientations of hyperedges. They also show that Theorem 3.1 for (not positively) intersecting supermodularity, follows from the theory of submodular flows using a reduction of Schrijver [24]. They indicate, however, that they do not know whether the polyhedron $P(D, f, \mathcal{F}, \mathcal{E}, l, u)$ for positively intersecting supermodular functions arises from a submodular flow polyhedron.

We proceed with a proof of Theorem 3.1. Henceforth we let $P(D, f, \mathcal{F}, \mathcal{E}, l, u)$ be a positively intersecting supermodular f-connectivity polyhedron and x^* be an extreme point. Note that any extreme point has a *defining system* determined by a triple $\mathcal{S}, \mathcal{R}^+, \mathcal{R}^-$, where $\mathcal{S} \subseteq \mathcal{F}$, $\mathcal{R} = \mathcal{R}^+ \cup \mathcal{R}^- \subseteq A \cup \mathcal{E}$, and $m = |A| = |\mathcal{S}| + |\mathcal{R}|$. That is, x^* is the unique solution (in R^A) to the system of equalities

1. $x_a = u_a$ for each $a \in \mathcal{R}^+$,
2. $x_a = l_a$ for each $a \in \mathcal{R}^-$,
3. $x_a + x_{a^-} = u_{a,a^-}$ for each $\{a, a^-\} \in \mathcal{R}^+$,
4. $x_a + x_{a^-} = l_{a,a^-}$ for each $\{a, a^-\} \in \mathcal{R}^-$,
5. $x(\delta^+(S)) = f(S)$ for each $S \in \mathcal{S}$,

and in particular, $\mathcal{S}, \mathcal{R}$ identify a set of linearly independent rows in the constraint matrix for the f-connectivity problem.

We now analyze the structure of such an extreme point x^* and this (not necessarily unique) defining system.

LEMMA 3.2. *If $(D, f, \mathcal{F}, \mathcal{E}, l, u)$ is a positively intersecting supermodular problem, then any extreme point x^* has a defining system such that $\mathcal{S}$ is laminar.*

Proof. We first produce a defining system which is cross-free. Suppose that $\mathcal{S}, \mathcal{R}$ gives a defining system and there exists $X, Y \in \mathcal{S}$ such that none of $X \cap Y, X - Y, Y - X, V - (X \cup Y)$ is empty. Let $x_{ab} = x^*([X - Y, Y - X])$, $x_{ba} = x^*([Y - X, X - Y])$, $x_{io} = x^*([X \cap Y, V - [X \cup Y])$, $x_{ia} = x^*([X \cap Y, X - Y])$, $x_{ib} = x^*([X \cap Y, Y - X])$, $x_{b0} = x^*([Y - X, V - (X \cup Y)])$, $x_{a0} = x^*([X - Y, V - (X \cup Y)])$. Then we have

$$\begin{aligned}
&(x_{ab} + x_{io} + x_{a0} + x_{ib}) + (x_{io} + x_{ba} + x_{b0} + x_{ia})\\
&= f(X) + f(Y)\\
&\leq f(X \cap Y) + f(X \cup Y)\\
&= (x_{ib} + x_{ia} + x_{io}) + (x_{io} + x_{b0} + x_{a0}),
\end{aligned}$$

from which we deduce that $X \cap Y, X \cup Y$ are also tight for x^*. In fact, we have $x_{ab} = x_{ba} = 0$, and hence the sum of the two constraints X, Y is identical to the sum of the constraints for $X \cap Y, X \cup Y$. Thus, replacing X, Y in $\mathcal{S}$ by $X \cap Y, X \cup Y$ we obtain another nonsingular constraint matrix and hence a defining system. Applying this procedure increases the value $\sum_{S \in \mathcal{S}} |S|^2$ and so we iteratively repeat this operation to obtain a cross-free family.

So we now assume that $\mathcal{S}$ is cross-free and suppose that $X, Y \in \mathcal{S}$ such that $X \cap Y, X - Y, Y - X$ are all nonempty but $X \cup Y = V$. Let $x_{ab} = x^*([X - Y, Y - X])$, $x_{ba} = x^*([Y - X, X - Y])$, $x_{ia} = x^*([X \cap Y, X - Y])$, $x_{ib} = x^*([X \cap Y, Y - X])$. For example, $x_{ia} = \sum_{a \in \delta^+(X \cap Y) - \delta^+(X)} x_a$; in particular, $x^*(X) = x_{ab} + x_{ib}, x^*(Y) = x_{ba} + x_{ia}$. Thus we have

$$\begin{aligned}
&(x_{ab} + x_{ib}) + (x_{ba} + x_{ia})\\
&= f(X) + f(Y)\\
&\leq f(X \cap Y) + f(X \cup Y)\\
&= f(X \cap Y)\\
&= (x_{ib} + x_{ia}),
\end{aligned}$$

from which we deduce that $x_{ab} = x_{ba} = 0$ and that $X \cap Y$ is again tight for x^*. In this case, we may replace the set Y in $\mathcal{S}$ by $X \cap Y$. Let a, b, c be the $0, 1$ incidence vectors of $\delta^+(X), \delta^+(Y), \delta^+(X \cap Y)$, respectively. Note that we have $a + c = 2a + b$ and hence the resulting system is again defining (i.e., induce a nonsingular matrix) for the vector x^*.

Applying this procedure decreases the value $\sum_{S \in \mathcal{S}} |S|$ and does not create any new intersecting pairs. Thus we may repeatedly apply this operation until we obtain the desired laminar system. □

Proof of Theorem 3.1. We now consider a basic solution x^* and a defining system $\mathcal{S}, \mathcal{R}$ for which $\mathcal{S}$ is laminar. We show that the constraint matrix associated with the inequalities for $\mathcal{S} \cup \mathcal{R}$ is a network matrix. Thus the defining system is determined by a totally unimodular matrix from which integrality of x^* follows. We construct a tree T, following Edmonds and Giles [3], as follows. For each $S \in \mathcal{S}$ create a node v_S. We also let v^* denote a "top node." Now for each maximal set in $\mathcal{S}$, we add an arc from

v^S to v^* with capacity equal to $f(S)$. For each other set S, let S' be the minimal set containing S. We then add an arc from v_S to $v_{S'}$ of capacity $f(S)$.

Now for each arc $a = (u, v)$ in D, let S (S') denote the minimal (maximal) set of $\mathcal{S}$ such that $a \in \delta^+(S)$. Let $v_a \in T$ be the head of the unique arc out of $v_{S'}$. Thus the arcs of the directed path P_a in T from v_S to v_a is in one-to-one correspondence with the cuts of $\mathcal{S}$ that contain a. Note also, that if there exists a^-, then it also has such a directed path P_{a^-} whose head is the same node v_a. Thus if $\{a, a^-\}$ is in the set $\mathcal{R}^+$, then we add a new node u_a and new arc (v_a, u_a) with the capacity u_{a,a^-}, accordingly. Similarly, for constraints in $\mathcal{R}^-$ (note that $\{a, a^-\}$ cannot be in both!). Finally, if a is in some upper or lower bound constraint, we may similarly add a new node w_a hanging from v_S with capacity l_a or u_a accordingly. This completes the definition of the tree T.

Finally, let D' be the digraph on $V(T)$ obtained as follows. For each arc a as above, we include an arc from v_S or w_a (depending on whether a is in a bound constraint) to v_a or u_a (depending on whether a is in some orientation constraint). It is clear that the defining system induced by $\mathcal{S} \cup \mathcal{R}$ is obtained from the network matrix for T, D' and right-hand sides given by capacities on the arcs in T. Since the resulting constraint matrix is totally unimodular, the proof is complete. □

4. Strong connectivity.

We present here a combinatorial 4-approximation algorithm for the problem of strong edge-connectivity with orientation constraints. (Henceforth, we drop the term "edge" and refer only to strong connectivity.) This is an important NP-hard special case of (I) that is not captured by our study in section 3—the requirement function for this problem is crossing supermodular. For clarity of presentation, we assume that any parallel arcs form a digon, and any such pair is involved in an orientation constraint; that is, the pair appears in $\mathcal{E}$. However, our algorithm extends trivially to the general case. In what follows, we assume that the input directed graph is $D = (V, A)$ and the optimal solution is a directed graph $D^* = (V, A^*)$. We use OPT to denote the cost of the optimal directed graph D^*. We say that an arc (u, v) in a directed graph $D = (V, A)$ is *simple* if $(v, u) \notin A$, and we say that it is *nonsimple* otherwise. A directed cycle is called *nontrivial* if it is a simple cycle of length at least three.

At the center of our approach is a procedure that takes as input any strongly connected subgraph of D, possibly violating some orientation constraints, and reduces the problem of "amending" its violated orientation constraints to that of finding a minimum cost 2-edge-connected subgraph in an undirected graph. In fact, the precise problem we reduce it to is a minimum cost augmentation of a spanning tree to a 2-edge-connected subgraph. For the latter problem, a 2-approximation algorithm [5, 17] is known. We now describe our algorithm in detail.

1. Pick any node r and compute a minimum cost in-branching to r, say, T_1, as well as a minimum cost out-branching from r, say, T_2. Consider the directed graph $D_1 = (V, A_1)$ induced by $T_1 \cup T_2$. Clearly, D_1 is strongly connected and its cost is at most $2 \cdot \mathrm{OPT}$. Assume without loss of generality that D_1 is minimal.
2. The set of simple arcs $A' \subseteq A_1$ induces a collection of strongly connected components $C_1, C_2, \ldots, C_k$ (see Lemma 4.1). Shrink each component C_i to a single node x_i and construct a directed graph $D_2 = (X, A_2)$, where $X = \{x_1, \ldots, x_k\}$. An arc $(x_i, x_j) \in A_2$ if and only if D_1 contained some arc (u, v) with $u \in X_i, v \in X_j$. The minimality of D_1 implies that D_2 is minimally strongly connected as well.

3. Replacing each nonsimple pair of arcs by an undirected edge evidently results in a tree by minimality. Thus, D_2 has $k-1$ nonsimple arc pairs $(a_1, b_1), \ldots, (a_{k-1}, b_{k-1})$. It is convenient to view D_2 as an undirected tree $T = (X, E_T)$, such that T contains an edge e_i for each pair (a_i, b_i). Let $(X_i, \bar{X}_i)$ be the partition of node set X induced by removal of a pair (a_i, b_i). Associate with any such partition a cut $(S_i, \bar{S}_i)$ in D_1, where $S_i = \bigcup_{x_j \in X_i} V(C_j)$. We refer to these cuts as the *fundamental cuts* of D_1. We say that an arc *a hits* a fundamental cut if $a \in \delta^+(S_i) \cup \delta^+(\bar{S}_i)$. For each fundamental cut, $A^* \setminus A_1$ contains an arc $a \in \delta^+(S_i) \cup \delta^+(\bar{S}_i)$ (see Lemma 4.2). Thus, the minimum cost Z of a set of arcs in $A \setminus A_2$ that hits all fundamental cuts of D_1 (call it the *fundamental directed cut cover* of D_1) is no more than OPT. Finding an optimal fundamental directed cut cover is NP-hard, but a 2-approximation can be easily obtained through the undirected version of the problem (see Lemma 4.3). Let A_3 be a set of arcs obtained in this fashion. Since Z is at most OPT, the cost of A_3 is at most $2 \cdot \mathrm{OPT}$. Let $D_3 = (V, A_1 \cup A_3)$ be the directed graph obtained by adding arcs in A_3 to the directed graph D_1. The total cost of arcs in D_3 is at most $4 \cdot \mathrm{OPT}$.
4. The final step is to show that the directed graph D_3 above can be modified into another directed graph $D_4 = (V, A_4)$ such that (i) D_4 is strongly connected, (ii) $A_4 \subseteq (A_2 \cup A_3)$, and (iii) all arcs in D_4 are simple. To achieve this we use the necessary and sufficient conditions for the existence of a strong orientation of a mixed graph given by Boesch and Tindell [2] (see Lemma 4.4). The costs of steps (1) and (3) are no more than $2 \cdot \mathrm{OPT}$ each. Therefore we now have a 4-approximation to our problem.

LEMMA 4.1. *In any minimally strongly connected directed graph H, the set of simple arcs induces a collection of strongly connected components.*

Proof. It suffices to show that every simple arc lies on a directed cycle that consists only of simple arcs. Consider any simple arc (u, v) and a path $P(v, u)$ from v to u in H. By the minimality of H, every arc on $P(v, u)$ must be simple. The lemma then follows. □

LEMMA 4.2. *$A^* \setminus A_1$ hits every fundamental cut of D_1.*

Proof. Suppose not. Then there is a cut $(S_i, \bar{S}_i)$ such that A^* has at most one arc (from the pair $\{a_i, b_i\}$) that crosses the cut. This contradicts that D^* is strongly connected. □

LEMMA 4.3. *There is a 2-approximation algorithm for the minimum cost fundamental directed cut cover problem.*

Proof. We solve this problem by a reduction to the minimum cost 2-edge-connected subgraph problem. Let B be the set of arcs $A \setminus A_2$. Consider the undirected graph $H = (X, E)$ obtained as follows. There is an edge $x_i x_j \in E$ if and only if there is an arc in B that connects some node in X_i to X_j or vice versa. Moreover, the cost of this edge is equal to the minimum cost such arc. Also, for each edge in E_T, we include an edge of cost 0 in H.

We now claim that the problem of finding a minimum cost fundamental directed cut cover of D_1 is equivalent to finding a minimum cost 2-edge-connected subgraph $H' = (X, E')$ of H. To see this, consider any fundamental directed cut cover $\hat{A}$. Then, the set of undirected edges in H, obtained from the arcs in $\hat{A}$ along with the edges in E_0, has the property that every cut in H has at least two edges crossing it. Moreover, the cost of this collection of edges is no more than the cost of the fundamental directed cut cover $\hat{A}$. In the other direction, let us consider a 2-edge-connected subgraph $\hat{E}$

of H. Since exactly one edge in E_0 crosses a cut in H, there must be at least one additional edge in $\hat{E}$ crossing any cut. Thus, the directed arcs corresponding to the edges in $\hat{E}$ form a fundamental cut cover of D_1.

Since the zero cost edges induce a spanning tree, the minimum cost 2-edge-connected subgraph problem we need to solve is essentially a minimum cost augmentation of a spanning tree to a 2-edge-connected subgraph. For this problem, a 2-approximation algorithm is known [5, 17]. It then suffices to use this algorithm to get a 2-approximation algorithm for the minimum cost fundamental directed cut cover problem. □

LEMMA 4.4. *The directed graph D_3 obtained at the end of step* (3) *of the algorithm can be modified into a directed graph D_4 such that D_4 is strongly connected and contained in D_3 and does not contain any nonsimple arcs.*

Proof. The lemma essentially follows from the necessary and sufficient conditions of Boesch and Tindell [2] for the existence of a strong orientation of a mixed graph. The Boesch–Tindell (BT) conditions state that a mixed graph whose underlying graph is 2-edge-connected has a strong orientation if and only if it does not contain a directed cut, i.e., a cut where all edges are directed in the same direction. Note that the underlying graph of D_3 is 2-edge-connected, and since A_2 spans D_3, every cut contains an undirected edge and therefore there are no directed cuts in D_3.

For completeness, we sketch here a short proof that does not use the BT conditions directly. The only nonsimple arcs (or undirected edges) in D_3 are the ones corresponding to the pairs (a_i, b_i). We remove from D_3, one by one, exactly one arc from each such pair while keeping it strongly connected. Consider a leaf x_i in T. There must be an arc in A_3 of the form (x_i, x_j) or (x_j, x_i) that hits the fundamental cut $(x_i, X \setminus \{x_i\})$. Assume without loss of generality that it is of the form (x_i, x_j). Consider the directed path from x_j to x_i in D_2, consisting only of nonsimple arcs, and remove every nonsimple arc that is oriented in a direction opposite to this path. Contract the resulting cycle and repeat this procedure on a leaf of the resulting tree T'. It is easy to verify that the procedure continues until the resulting tree reduces to a single node, corresponding to the graph D_4 above. □

5. Crossing supermodularity.

We now focus our attention on general crossing supermodular functions. While simple constant factor approximation algorithms are known in the absence of orientation constraints [22], the problem seems to become much harder in the presence of orientation constraints. Although we do not resolve this question here, we make some progress toward solving our original problem (I) for crossing supermodular functions. We establish the following result.

THEOREM 5.1. *Let* OPT *denote the optimal cost of a fractional solution to the problem* (I) *with a crossing supermodular function f defined on a crossing family $\mathcal{F}$. Let $p = \max_S f(S)$ denote the maximum requirement of any set. Then, there is a polynomial time algorithm that finds an integral solution of cost $2p \cdot \mathrm{OPT}$ satisfying the weaker requirement function f' defined as $f'(S) = \max\{f(S) - 1, 0\}$.*

An immediate corollary of the above theorem is that we can find a solution to a $(k-1)$-strong edge-connectivity problem with orientation constraints at a cost that is no more than $2k$ times the optimal cost for k-strong edge-connectivity. We devote the rest of this section to the proof of Theorem 5.1.

A key step in our algorithm finds a minimum cost orientation of a graph for general crossing supermodular requirement functions. That is, it solves problem (I) in the case where the orientation constraints are equalities. As mentioned earlier, this problem can be solved in polynomial time via reductions to submodular flows; see

[9, 24]. (In [10] an orientation theorem is given for the more general class of so-called crossing G-supermodular functions.) We briefly explain how this is done, following [9].

First, choose the cheaper arc from each pair of arcs, a and a^-, appearing in an orientation constraint $x_a + x_{a^-} = 1$, yielding a directed graph denoted by $D_1 = (V, A_1)$. Clearly, the cost of A_1 is a lower bound on the cost of an optimal solution. Then, find a minimum cost collection of arcs $(u, v) \in A_1$, such that if they are *flipped*, i.e., (u, v) is replaced by (v, u), then a directed graph satisfying the requirement function f is obtained. Call such a set of arcs an *f-flip set.*

Minimum cost f-flip sets can be optimally computed by a reduction to the minimum cost submodular flow problem, which is itself solvable in polynomial time. It is easily seen that a set A_2 is an f-flip set if and only if its incidence vector x satisfies, for every proper subset S,

$$|\delta^+(S)| + x(\delta^-(S)) - x(\delta^+(S)) \geq f(S).$$

Define a new function g where

$$g(S) = -f(S) + |\delta^+(S)|.$$

Since f is crossing supermodular, we get that g is submodular. Therefore, finding a minimum cost f-flip set is equivalent to solving the submodular flow problem $\min\{cx : x \geq 0,\ x(\delta^+(S)) - x(\delta^-(S)) \leq g(S)$ for each proper subset $S\}$.

We now go back to the proof of Theorem 5.1. Let D be our digraph and f be a crossing supermodular function on the family $\mathcal{F}$. Let $p = \max_S(f(S) - 1)$. Define $f'(S) = \lfloor \frac{f(S)}{1+1/p} \rfloor$ for each set S. It is easy to verify that $f'(S) = f(S) - 1$ if $f(S) > 0$, and that f' is also crossing supermodular on $\mathcal{F}$. Let $\mathrm{OPT}(f)$, or OPT if the context is clear, denote the cost of an optimal solution x^* to problem (I) for a requirement function f.

1. Define u_a to be $\lceil px^*(a) \rceil$. Clearly, $u_a + u_{a^-} \leq p + 1$. Also, define $f_p(S) = p \cdot f(S)$ to be a new crossing supermodular function.
2. We now solve two separate intersecting supermodular LPs, corresponding to f_p, with upper bounds just defined. These LPs are obtained by splitting the "requiring" sets into $\mathcal{F}^1 = \{F \in \mathcal{F} : v \in F\}$ and $\mathcal{F}^2 = \mathcal{F} - \mathcal{F}^1$ where v is an arbitrarily chosen node. The first we may solve directly; the latter is not actually intersecting and so we work with the intersecting family $\{V - S : S \in \mathcal{F}^2\}$. For the second family we must also work with the function f^* defined by $f^*(S) = f(V - S)$ and use the digraph with the arcs reversed (see [22]). By Theorem 3.1, any basic solution for these LPs is integral. We may thus find two such vectors z^1, z^2 in polynomial time. Actually, since these LPs do not have orientation constraints, we can also find integral optimal solutions using Frank's approach ([6]; see also [24]). Now define a new vector z by setting $z_a = \max\{z_a^1, z_a^2\}$ for each $a \in A$. Clearly z is integral, satisfies f_p, and costs no more than $2 \cdot \mathrm{OPT}(f_p) \leq 2p \cdot \mathrm{OPT}(f)$.
3. By setting $y = \frac{1}{p}z$ we obtain a solution for the original function f which is $(1/p)$-integral and has cost at most 2OPT. The solution y violates any orientation constraint by at most a factor of $1 + 1/p$. We scale down all violating arc pairs to satisfy the constraint $x_a + x_{a^-} = 1$. We also uniformly scale up any arc pairs with $1/p \leq x_a + x_{a^-} < 1$ to satisfy $x_a + x_{a^-} = 1$. The resulting solution clearly satisfies the function f' and has a cost that is at most $2p \cdot \mathrm{OPT}$.

4. At this point, since all orientation constraints are tight, we can find an integral solution of no greater cost.

We can view the above algorithm as a process in which we tighten the orientation constraints, while bounding the increase in cost. However, we can guarantee only that the cut constraints are "almost" satisfied. We conjecture that the following approach, similar to that of Jain [15] and Melkonian and Tardos [22] can yield a constant factor approximation algorithm for crossing supermodular requirement functions. Let x^* be an optimal (basic) solution to the linear relaxation of (I) in the case where f is crossing supermodular. We conjecture that there exists a pair of arcs $a, a^- \in A$, for which the orientation constraint is not tight, yet $x^*(a) + x^*(a^-)$ is greater than a constant, say, 1/4. If this conjecture is correct, then we can make the orientation constraint on a and a^- tight and resolve the problem. We repeat this until we obtain a solution in which all orientation constraints are tight. The cost of this solution increases by only a constant factor with respect to x^*. Once all orientation constraints are tight, as before, we can find an integral solution at no greater cost.

Acknowledgments. The authors wish to thank the referees for helpful comments which greatly improved the paper and also Joseph Cheriyan who made a number of important remarks. The authors are very grateful to Andras Frank for many thoughtful and detailed comments on the contents of the paper.

REFERENCES

[1] A. Agrawal, P. Klein, and R. Ravi, *An approximation algorithm for generalized Steiner tree problems in networks*, in Proceedings of the 23rd ACM Symposium on Theory of Computing, 1991, pp. 134–144.

[2] F. Boesch and R. Tindell, *Robbins's theorem for mixed multigraphs*, Amer. Math. Monthly, 87 (1980), pp. 716–719.

[3] J. Edmonds and R. Giles, *A min-max relation for submodular functions on graphs*, Ann. Discrete Math., 1 (1977), pp. 185–204.

[4] K. P. Eswaran and R. E. Tarjan, *Augmentation problems*, SIAM J. Comput., 5 (1976), pp. 653–665.

[5] G. N. Frederickson and J. JáJá, *Approximation algorithms for several graph augmentation problems*, SIAM J. Comput., 10 (1981), pp. 270–283.

[6] A. Frank, *Kernel systems of directed graphs*, Acta Sci. Math (Szeged), 41 (1979), pp. 63–76.

[7] A. Frank, *How to make a digraph strongly connected*, Combinatorica, 1 (1981), pp. 145–153.

[8] A. Frank, *Augmenting graphs to meet edge-connectivity requirements*, SIAM J. Discrete Math., 5 (1992), pp. 25–53.

[9] A. Frank, *Applications of submodular functions*, in Surveys in Combinatorics, London Math. Soc. Lecture Note Ser. 187, K. Walker, ed., Cambridge University Press, Cambridge, UK, 1993, pp. 85–136.

[10] A. Frank, *Orientations of graphs and submodular flows*, Congr. Numer., 113 (1996), pp. 111–142.

[11] A. Frank, T. Király, and Z. Király, *On the Orientation of Graphs and Hypergraphs*, Technical Report TR-2001-06, Egerváry Research Group, Budapest, Hungary, 2001.

[12] M. Goemans and D. Williamson, *A general approximation technique for constrained forest problems*, SIAM J. Comput., 24 (1995), pp. 296–317.

[13] M. Goemans, A. Goldberg, S. Plotkin, D. Shmoys, É. Tardos, and D. Williamson, *Approximation algorithms for network design problems*, in Proceedings of the 5th Annual ACM–SIAM Symposium on Discrete Algorithms, 1994, pp. 223–232.

[14] M. Grötschel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica, 1 (1981), pp. 169–197.

[15] K. Jain, *A factor* 2 *approximation algorithm for the generalized Steiner network problem*, Combinatorica, 21 (2001), pp. 39–60.

[16] S. Khanna, J. Naor, and F. B. Shepherd, *Directed network design with orientation constraints*, in Proceedings of the 11th Annual ACM–SIAM Symposium on Discrete Algorithms, 2000, pp. 663–671.

[17] S. Khuller and R. Thurimella, *Approximation algorithms for graph augmentation*, J. Algorithms, 14 (1993), pp. 214–225.

[18] T. Király, *Edge-Connectivity of Undirected and Directed Hypergraphs*, Ph.D. thesis, Eötvös University, Budapest, Hungary, 2003.

[19] L. Lovász, *On two min-max theorems in graph theory*, J. Comput. Theory, 21 (1976), pp. 96–103.

[20] L. Lovász, *Combinatorial Problems and Exercises*, North-Holland Press, Amsterdam, 1979.

[21] C. L. Lucchesi and D. H. Younger, *A minimax relation for directed graphs*, J. London Math. Soc., 17 (1978), pp. 369–374.

[22] V. Melkonian and É. Tardos, *Approximation algorithms for a directed network design problem*, in Proceedings of the 7th International Integer Programming and Combinatorial Optimization Conference, Lecture Notes in Comput. Sci. 1610, Springer-Verlag, Berlin, 1999, pp. 345–360.

[23] A. Schrijver, *Packing and covering of crossing families of cuts*, J. Combin. Theory, 35 (1983), pp. 104–128.

[24] A. Schrijver, *Total dual integrality from directed graphs, crossing families and sub- and supermodular functions*, in Progress in Combinatorial Optimization, W. R. Pulleyblank, ed., Academic Press, Toronto, 1984, pp. 315–362.

[25] D. Williamson, M. Goemans, M. Mihail, and V. Vazirani, *A primal-dual approximation algorithm for generalized Steiner network problems*, Combinatorica, 15 (1995), pp. 435–454.

[26] D. Younger, *private communication*, 1985.

SIAM J. Discrete Math.
Vol. 19, No. 1, pp. 258–280

ON THE POWER OF NONLINEAR SECRET-SHARING[*]

AMOS BEIMEL[†] AND YUVAL ISHAI[‡]

Abstract. A *secret-sharing scheme* enables a dealer to distribute a secret among n parties such that only some predefined authorized sets of parties will be able to reconstruct the secret from their shares. The (monotone) collection of authorized sets is called an *access structure*, and is freely identified with its characteristic monotone function $f : \{0,1\}^n \to \{0,1\}$. A family of secret-sharing schemes is called *efficient* if the total length of the n shares is polynomial in n. Most previously known secret-sharing schemes belonged to a class of *linear* schemes, whose complexity coincides with the *monotone span program* size of their access structure. Prior to this work there was no evidence that nonlinear schemes can be significantly more efficient than linear schemes, and in particular there were no candidates for schemes efficiently realizing access structures which do not lie in NC.

The main contribution of this work is the construction of two efficient nonlinear schemes: (1) A scheme with perfect privacy whose access structure is conjectured not to lie in NC, and (2) a scheme with statistical privacy whose access structure is conjectured not to lie in P/poly. Another contribution is the study of a class of nonlinear schemes, termed *quasi-linear* schemes, obtained by *composing* linear schemes over different fields. While these schemes are (superpolynomially) more powerful than linear schemes, we show that they cannot efficiently realize access structures outside NC.

Key words. secret-sharing, nonlinear secret-sharing, monotone span programs, quadratic residuosity

AMS subject classifications. 68Q99, 68R05, 94A60, 94A62

DOI. 10.1137/S0895480102412868

1. Introduction.

Secret-sharing schemes enable a dealer, holding a secret piece of information, to distribute this secret among n parties such that only some predefined authorized sets of parties can reconstruct the secret from their shares and others learn nothing about it. The (monotone) collection of authorized sets that can reconstruct the secret is called an *access structure*, and is freely identified with its characteristic monotone function $f : \{0,1\}^n \to \{0,1\}$.

The first secret-sharing schemes were introduced by Blakley [14] and Shamir [52]. They constructed *threshold* schemes, in which the access structure is defined by a threshold function. General secret-sharing schemes, realizing nonthreshold access structures, were introduced by Ito, Saito, and Nishizeki [43], where it was shown that every monotone access structure can be (inefficiently) realized by a secret-sharing scheme. More efficient schemes for specific types of access structures were presented, e.g., in [10, 54, 18, 45]. We refer the reader to [53, 58] for extensive surveys on secret-sharing literature.[1]

[*] Received by the editors August 14, 2002; accepted for publication (in revised form) December 6, 2004; published electronically September 8, 2005. A preliminary version of this paper appeared in the proceedings of the 16th Annual IEEE Conference on Computational Complexity, Chicago, IL, 2001, pp. 188–202.

http://www.siam.org/journals/sidma/19-1/41286.html

[†] Computer Science Department, Ben-Gurion University, Beer-Sheva 84105, Israel (beimel@cs.bgu.ac.il).

[‡] Computer Science Department, Technion, Haifa 32000, Israel (yuvali@cs.technion.ac.il). The work of this author was done while at AT&T Labs–Research and DIMACS.

[1] Similar to almost all of the vast literature on secret-sharing, this work is concerned with the *information-theoretic* variant of the problem. A relaxed notion of *computationally* secure secret-sharing has been considered in [62, 46, 4].

Originally motivated by the problem of secure information storage, secret-sharing schemes have found numerous other applications in cryptography and distributed computing (see [50, 9, 23, 27, 30]). However, secret-sharing is independently interesting as a pure complexity question. The default complexity measure of secret-sharing schemes is their *share size*, i.e., the total length of all shares distributed by the dealer. This is a measure of the amount of communication (or storage) required for sharing a secret.[2] One of the most interesting open questions in this area is to characterize which access structures can be *efficiently* realized, i.e., with shares of polynomial size in the number of parties n. For most access structures, the best known upper bound on the share size is exponential. However, unlike other concrete complexity measures such as circuit complexity, one cannot apply simple counting arguments to show that this must indeed be the case. In fact, given the current knowledge, one cannot even rule out the possibility that *all* access structures can be efficiently realized.

Several lower bounds on the share size of secret-sharing were obtained [22, 15, 32, 29, 28]. The strongest current bound is $\Omega(n^2/\log n)$ [28]. This bound applies to an *explicit* access structure. However, as noted above, there is a huge gap between these lower bounds and the best known upper bounds.

1.1. Linear vs. nonlinear secret-sharing.

Most previously known secret-sharing schemes were *linear*. In a linear scheme, the secret is viewed as an element of a finite field F, and the shares are obtained by applying a linear mapping to the secret and several independent random field elements. Linear schemes may be equivalently defined by requiring that each authorized set reconstructs the secret by applying a linear function to its shares [6]. For example, the schemes of [52, 14, 43, 10, 54, 18, 13, 45, 33] are all linear.

The share size in linear schemes over F realizing a monotone function f is proportional to the *monotone span program* size of f over F. (Span programs are a linear-algebraic model of computation introduced in [45].) In fact, there is a one-to-one correspondence between linear secret-sharing schemes and monotone span programs. The class of functions that have polynomial-size monotone span programs, which coincides with those admitting efficient linear secret-sharing schemes, is fairly well understood: (1) it contains monotone NC^1 and even monotone symmetric logspace [10, 11, 45]; (2) it is contained in algebraic NC^2 (as follows from [12, 17, 49, 21]), implying that it is contained in NC^3 when $\log|F|$ is polynomially bounded; and (3) there are explicit monotone functions that are provably not in this class [7, 2, 37] (this is proved without any complexity assumptions).

As opposed to linear secret-sharing schemes, nearly nothing is known for general (i.e., possibly nonlinear) schemes. Several constructions of nonlinear secret-sharing schemes have been suggested, both for the threshold case [61, 31, 51] and for general access structures [35].[3] The question of basing verifiable secret-sharing and secure multiparty computation on nonlinear secret-sharing has been studied in [26]. However, none of these works provide evidence that nonlinear schemes are significantly more powerful than their linear counterparts.

The relation between linear and nonlinear complexity has been studied in other contexts, such as coding and randomness extraction (see [60]). While in some of these contexts the margins of possible improvement obtained by relaxing the linearity restriction are provably small, this is not the case for our problem. As discussed

[2] By default, we ignore the *computational* complexity of the scheme. However, most of our efficient constructions are also computationally efficient. We explicitly indicate when this is not the case.

[3] A nonlinear construction of [19] has been shown to be incorrect by [55].

above, it is not even known if there exists an access structure that *cannot* be efficiently realized by a nonlinear scheme. On the other hand, prior to this work there was no evidence that nonlinear schemes are significantly more efficient than linear schemes. In particular, there were no explicit candidates for secret-sharing schemes realizing access structures which do not lie in NC.

1.2. Our results. We attempt to remedy the above state of affairs. To this end, we take two different approaches.

Specific candidates. The main contribution of this work is the construction of specific efficient nonlinear secret-sharing schemes, whose access structures are conjectured to be hard. We present two main schemes, whose access structures are related to two variants of the quadratic residuosity problem.[4] A third scheme, which is a simplified version of the second, realizes an access structure related to the coprimality problem.[5]

The first scheme realizes an access structure whose computational complexity is equivalent to that of deciding quadratic residuosity modulo a *fixed* prime, where the prime modulus may depend only on the number of parties.[6] This problem is not known to be in NC. In particular, assuming that it is indeed not in NC, a separation of efficient nonlinear schemes from efficient linear schemes follows.

The second scheme realizes a presumably much harder access structure, whose computational complexity is equivalent to the general quadratic residuosity problem. The latter is widely conjectured to require superpolynomial (or even exponential) size circuits, and its intractability is implied by the so-called *quadratic residuosity assumption* [39], which is commonly relied on in cryptography. In contrast to the first construction, the second construction only meets a more liberal notion of secret-sharing (with a statistical relaxation of the perfect correctness and privacy requirements; see section 2), and its reconstruction procedure is computationally inefficient. Yet, the second scheme demonstrates that the share size in a secret-sharing scheme may be superpolynomially smaller than the circuit size of its access structure.

As a variant of the second scheme described above, we obtain a scheme whose access structure is equivalent to the coprimality problem. Similarly to quadratic residuosity modulo a (fixed) prime, the coprimality problem is in P but is not known to be in NC. As the second scheme, the third scheme meets only the more liberal notion of security. However, unlike the second scheme it is also computationally efficient. Compared to the first scheme, the coprimality problem is more standard than the problem of deciding quadratic residuosity modulo a *fixed* prime. The main properties of the three schemes described above are summarized in Table 1.1.

Our constructions were inspired by a noninteractive private protocol for the quadratic residuosity problem from [36]. In fact, every protocol in the model of [36, 42] can be transformed into a secret-sharing scheme for a related access structure.

Quasi-linear schemes. In addition to the above specific candidates, we study a class of nonlinear schemes, which we term *quasi-linear* schemes, obtained by *composing* linear schemes over (possibly) different fields. Composition of secret-sharing schemes has been used in previous works (see [10, 20, 59, 47, 27]). However, to the best of our

[4] The quadratic residuosity problem is that of deciding, given a pair of integers w, u, whether w is a square modulo u.

[5] The coprimality problem is that of deciding, given w, u, whether $\gcd(w, u) = 1$.

[6] While a generalization to quadratic residuosity modulo a *fixed* composite is possible, this problem is essentially equivalent in a nonuniform setting to deciding quadratic residuosity modulo a fixed prime.

TABLE 1.1
Summary of our main schemes.

	perfect/ statistical	access structure related to...	computat. efficient?	Hardness of access structure
section 3	perfect	quadratic residuosity modulo a fixed prime	yes	in P not known to be in NC
section 4	statistical	quadratic residuosity	no	in NP conjectured not in P/poly
section 4.2	statistical	coprimality	yes	in P not known to be in NC

knowledge this is the first work to explicitly discuss compositions of linear schemes over different fields. We characterize the complexity of quasi-linear schemes in terms of Boolean formulas over the basis of monotone span programs. We prove that quasi-linear schemes cannot realize any access structure outside NC. Specifically, we show that the class of structures which they can efficiently realize is contained in NC^4. Thus, quasi-linear schemes do not provide the strong (conjectured) results implied by the specific candidates described above. On a positive note, we show an application of quasi-linear schemes for the construction of secret-sharing schemes efficiently realizing monotone span programs over a ring $\mathcal{Z}_u$, where u is a square-free composite. A naive generalization of the construction for monotone span programs over *fields* [45] fails to achieve this goal.[7] Following our work, it was shown in [8] that quasi-linear schemes are strictly more powerful than linear schemes, that is, there are explicit functions that have small quasi-linear schemes; however, they require superpolynomial linear schemes.

Organization. In section 2 we present some definitions and background. In sections 3 and 4 we describe our two main constructions of efficient nonlinear schemes, and discuss the complexity of their access structures. Finally, in section 5 we introduce and study the class of quasi-linear schemes.

2. Preliminaries.

In this section we define secret-sharing schemes, linear secret-sharing schemes, and span programs, and briefly discuss the connections between these notions. We end this section with some definitions related to the quadratic residuosity problem.

DEFINITION 2.1 (access structure). *Let $\{P_0, \ldots, P_{n-1}\}$ be a set of parties. A collection $\mathcal{A} \subseteq 2^{\{P_0,\ldots,P_{n-1}\}}$ is* monotone *if $B \in \mathcal{A}$ and $B \subseteq C$ imply $C \in \mathcal{A}$. An* access structure *is a monotone collection $\mathcal{A}$ of nonempty subsets of $\{P_0, \ldots, P_{n-1}\}$ (that is, $\mathcal{A} \subseteq 2^{\{P_0,\ldots,P_{n-1}\}} \setminus \{\emptyset\}$). The sets in $\mathcal{A}$ are called the* authorized sets. *A set B is called a* minimal set *of $\mathcal{A}$ if $B \in \mathcal{A}$, and $C \notin \mathcal{A}$ for every $C \subsetneq B$. The minimal sets of an access structure uniquely define it. Finally, we freely identify an access structure with its monotone characteristic function $f_{\mathcal{A}} : \{0,1\}^n \to \{0,1\}$, whose variables are denoted $x_0, \ldots, x_{n-1}$.*

DEFINITION 2.2 (secret-sharing). *Let S be a finite set of secrets, where $|S| \geq 2$. An n-party* secret-sharing scheme *Π with* secret-domain *S is a randomized mapping from S to a set of n-tuples $S_0 \times S_1 \times \cdots \times S_{n-1}$, where S_i is called the* share-domain *of P_i. A dealer distributes a secret $s \in S$ according to Π by first sampling a vector of* shares *$(s_0, \ldots, s_{n-1})$ from $\Pi(s)$, and then privately communicating each share s_i to*

[7] This result does not follow from [35], which imposes stronger requirements in their definition of span programs over rings.

the party P_i. We say that Π realizes *an access structure* $\mathcal{A} \subseteq 2^{\{P_0,\ldots,P_{n-1}\}}$ *(or the corresponding monotone function* $f_{\mathcal{A}} : \{0,1\}^n \to \{0,1\}$*) if the following two requirements hold:*

Correctness. *The secret s can be reconstructed by any authorized set of parties. That is, for any set $B \in \mathcal{A}$ (where $B = \{P_{i_1}, \ldots, P_{i_{|B|}}\}$), there exists a* reconstruction function $\mathrm{Rec}_B : S_{i_1} \times \cdots \times S_{i_{|B|}} \to S$ *such that for every $s \in S$,*

$$\Pr[\, \mathrm{Rec}_B(\Pi(s)_B) = s \,] = 1,$$

where $\Pi(s)_B$ denotes the restriction of $\Pi(s)$ to its B-entries.

Privacy. *Every unauthorized set cannot learn anything about the secret (in the information theoretic sense) from their shares. Formally, for any set $C \notin \mathcal{A}$, for every two secrets $a, b \in S$, and for every possible shares $\langle s_i \rangle_{P_i \in C}$:*

$$\Pr[\, \Pi(a)_C = \langle s_i \rangle_{P_i \in C} \,] \quad = \quad \Pr[\, \Pi(b)_C = \langle s_i \rangle_{P_i \in C} \,].$$

The share complexity *of the scheme (or* complexity *for short) is defined as*

$$\sum_{i=0}^{n-1} \log |S_i|.$$

As the mutual information between the secret and the shares of a set of parties can only grow when we add parties to the set, it suffices to prove the correctness for minimal authorized sets and the privacy for maximal unauthorized sets.

The above correctness and privacy requirements capture the strict notion of *perfect* secret-sharing, which is the one most commonly referred to in the secret-sharing literature. We will also consider a relaxed but natural notion of *statistical* secret-sharing, in which Π accepts an additional argument k, called the *security parameter*, and the perfect correctness and privacy requirements are relaxed to *statistical correctness* and *statistical privacy* defined as follows:

Statistical correctness. *Any authorized set of parties can reconstruct the secret s except with* negligible *probability $\epsilon(k)$. That is, for every authorized $B \in \mathcal{A}$ there exists a reconstruction function* Rec_B *such that*

$$\Pr[\, \mathrm{Rec}_B(\Pi(s)_B) = s \,] \geq 1 - \epsilon(k) \tag{2.1}$$

for some $\epsilon(k) \in k^{-\omega(1)}$.

Statistical privacy. *Any unauthorized set of parties learns only a negligible amount of information about the secret. That is, for any unauthorized $C \notin \mathcal{A}$ and two secrets $a, b \in S$,*

$$\mathrm{SD}(\Pi(a,k)_C, \Pi(b,k)_C) \leq \epsilon(k) \tag{2.2}$$

for some $\epsilon(k) \in k^{-\omega(1)}$, where $\mathrm{SD}(Y_0, Y_1)$ *denotes the* statistical distance *between distributions Y_0, Y_1 defined as*

$$\mathrm{SD}(Y_0, Y_1) = \frac{1}{2} \sum_y |\Pr[Y_0 = y] - \Pr[Y_1 = y]|.^{8}$$

[8] Equivalently, the statistical distance between Y_0 and Y_1 may be defined as the maximum, over all functions A, of the *distinguishing advantage* $|\Pr[A(Y_0) = 1] - \Pr[A(Y_1) = 1]|$.

We next define the class of *linear* secret-sharing schemes. There are several equivalent definitions for these schemes; see [6].

DEFINITION 2.3 (linear secret-sharing). *Let F be a finite field. A secret-sharing scheme Π is said to be* linear *over F if the following hold:*

1. *The secret-domain S is a subset of F.*
2. *There exist $d_0, \ldots, d_{n-1}$ such that each share-domain S_i is a subspace of the vector space F^{d_i}.*
3. *The randomized mapping Π can be computed as follows. First, the dealer chooses independent random variables, denoted $r_1, \ldots, r_\ell$, each uniformly distributed over F. Then, each coordinate of each of the n shares is obtained by taking a* linear combination *of $r_1, \ldots, r_\ell$ and the secret s.*

We remark that the notions of perfect secret-sharing and statistical secret-sharing coincide in the case of linear schemes: Any linear scheme that satisfies the weaker conditions of statistical correctness and privacy satisfies the stronger requirements of perfect correctness and privacy.

Remark 2.4. Van Dijk [32] describes a generalization of linear schemes which, following [55], we call multilinear schemes. In a multilinear scheme the secret is viewed as a collection of elements from a finite field F, and the shares are obtained by applying a linear mapping to the elements of the secret and several independent random field elements. Simonis and Ashikhmin [55] show that multilinear schemes can be somewhat more efficient than linear schemes. However, if we require that the length of the secret is polynomial in the number of parties, then multilinear schemes can only be polynomially more efficient than linear schemes.

As for any other concrete complexity measure, we will often implicitly use the term "scheme" for referring to an infinite *family* of schemes $\{\Pi_n\}_{n \in \mathcal{N}}$, parameterized by the number of parties n. In the statistical case, we require the same negligible function $\epsilon(k)$ to apply in (2.1) and (2.2) for all Π_n in the family. In the linear case, such a family can have a different underlying field for each n. A family $\{\Pi_n\}_{n \in \mathcal{N}}$ is *efficient* if the complexity of Π_n is polynomial in n (or the complexity of $\Pi_n(k)$ is polynomial in n and k in the statistical case). Note that the above definition does not make any requirement on the computational complexity of the scheme. We say that the scheme is *computationally efficient* if both sharing the secret and reconstructing it can be done in time poly(n,k,log $|S|$). Finally, the family of access structures $\{\mathcal{A}_n\}$ realized by a scheme family $\{\Pi_n\}$ is naturally identified with a monotone Boolean function $f : \{0,1\}^* \to \{0,1\}$ or its characteristic language.

We next define span programs—a linear algebraic model of computation whose monotone version is equivalent to linear secret-sharing.

DEFINITION 2.5 (span program [45]). *A* span program *over a field F is a triplet $\widehat{M} = \langle M, \rho, \vec{v} \rangle$, where M is an $r \times c$ matrix over F, the vector $\vec{v} \in F^c$ is a nonzero row vector called the* target vector*, and ρ is a labeling of the rows of M by literals from $\{x_0, \bar{x}_0, \ldots, x_{n-1}, \bar{x}_{n-1}\}$ (every row is labeled by one literal, and the same literal can label many rows). A span program $\widehat{M}$ is said to be* monotone *if all of its rows are labeled by positive literals.*

A span program accepts or rejects an input by the following criterion. For every input $y \in \{0,1\}^n$ let M_y denote the submatrix of M consisting of those rows whose labels are satisfied by the assignment y. The span program $\widehat{M}$ accepts *y iff $\vec{v}$ is in the row-span of M_y (where each row of M is viewed as a vector in F^c). A span program* computes *a Boolean function $f : \{0,1\}^n \to \{0,1\}$ if it accepts exactly those inputs y such that $f(y) = 1$. Note that monotone span programs compute monotone functions.*

Finally, the size *of* $\widehat{M}$ *is the number of* rows *in* M.

The complexity of realizing a given access structure by a linear secret-sharing scheme over F is proportional to the minimal size of a monotone span program over F computing f. Specifically we refer to the following lemma.

LEMMA 2.6 (see [45, 6]). *An access structure can be realized by a linear secret-sharing scheme over* F *in which the shares include a total of* d *field elements iff it can be computed by a monotone span program over* F *of size* d.

It follows from [12, 17, 49, 21] that all functions that have small span programs are in NC. Specifically, we have the following.

LEMMA 2.7. *If a function* f *has a span program over* $F = \mathrm{GF}(q)$ *of size* ℓ*, then* f *has an arithmetic circuit of size* $\mathrm{poly}(\ell)$ *and depth* $O(\log^2 \ell)$ *over* F*, implying that it has a Boolean circuit of size* $\mathrm{poly}(\ell, \log q)$ *and depth* $O(\log^2 \ell \log\log q)$.

Quadratic residues. Let $\mathcal{Z}_u$ be the ring of integers modulo u, whose elements are identified with the integers $\{0, 1, \dots, u-1\}$. Let $\mathcal{Z}_u^*$ denote the multiplicative group of the elements of $\mathcal{Z}_u$ that are relatively prime to u, that is, the elements of $\mathcal{Z}_u^*$ are $\{1 \le w < u \ : \ \gcd(w, u) = 1\}$. The number of element in $\mathcal{Z}_u^*$ is denoted by $\varphi(u)$, and is referred to as the Euler function of u.

An integer w is said to be a *quadratic residue* modulo u if $\gcd(w, u) = 1$ and there exists an integer b such that $w \equiv b^2 \bmod u$. It is said to be a *quadratic nonresidue* modulo u if $\gcd(w, u) = 1$ and there is no integer b such that $w \equiv b^2 \bmod u$. We will pay particular attention to the case where the modulus is an odd prime p; thus, w and b may be viewed as elements of the field $\mathcal{Z}_p$. In this case, $w \in \mathcal{Z}_p^* = \mathcal{Z}_p \setminus \{0\}$ is said to be a quadratic residue if it is a square of some field element, and a *quadratic nonresidue* otherwise. (The element 0 is neither a quadratic residue nor a quadratic nonresidue.) The quadratic residues form a subgroup of the multiplicative group $\mathcal{Z}_p^*$. The *quadratic residuosity problem* is that of deciding, given w and u, whether w is a quadratic residue modulo u. When u is restricted to be a prime (or given the factorization of u) this problem can be solved in polynomial time, but is not known to have an efficient *parallel* algorithm. When u is arbitrary, this problem is widely assumed to be intractable; see section 3.1 for more details.

3. An efficient nonlinear scheme: The perfect case.

In this section we construct an efficient nonlinear secret-sharing scheme whose access structure is conjectured not to lie in NC. The scheme constructed in this section is *perfectly* private and correct. A *statistical* scheme realizing a computationally harder access structure will be given in the next section.

DEFINITION 3.1 (the nonquadratic residue modulo prime access structure ($\mathbf{NQRP}_p$)). *Let* p *be an odd prime and* $m \overset{def}{=} \lfloor \log p \rfloor$. *We define the* n*-party access structure* $\mathbf{NQRP}_p$*, where* $n \overset{def}{=} 2m$*, by specifying its collection of minimal sets. The parties of the access structure are denoted by* P_i^b*, where* $0 \le i < m$ *and* $b \in \{0, 1\}$. *With each* $w \in \{0,1\}^m$ *(also viewed as an* m*-bit integer) we naturally associate a set* B_w *of size* m *defined by* $B_w \overset{def}{=} \{P_i^{w_i} : 0 \le i < m\}$. *A set* B *is a minimal set of* $\mathbf{NQRP}_p$ *if the following hold:*

- $B = \{P_i^0, P_i^1\}$ *for some* $0 \le i < m$*, or*
- $B = B_w$ *for some* w *such that* w *is* not *a quadratic residue modulo* p. *(That is, it is either* 0 *or a quadratic nonresidue.)*

TABLE 3.1
A secret-sharing scheme for $\mathbf{NQRP}_p$.

	$s=0$	$s=1$
P_0^b, where $b \in \{0,1\}$	r^2+z_0	br^2+z_0
P_i^b, where $1 \le i < m,\ b \in \{0,1\}$	z_i	$2^i br^2 + z_i$

We let **NQRP** *denote a family of access structures such that the nth structure is* $\mathbf{NQRP}_p$ *for some p such that* $\lfloor \log p \rfloor = \lfloor n/2 \rfloor$ *(say, the least such p).*[9]

We next construct a secret-sharing scheme for **NQRP**.

THEOREM 3.2. *For every odd prime p there exists a perfect secret-sharing scheme for* $\mathbf{NQRP}_p$ *in which the secret-domain is* $\{0,1\}$ *and the share-domain of each party is* $\mathcal{Z}_p$.

Proof. We prove this theorem by describing the secret-sharing scheme.

The dealer chooses at random $m-1$ random elements $z_0, z_1, \ldots, z_{m-2} \in \mathcal{Z}_p$ and an additional random element $r \in \mathcal{Z}_p^*$. Define

$$z_{m-1} \stackrel{\text{def}}{=} -\sum_{i=0}^{m-2} z_i, \tag{3.1}$$

where here and in the following all arithmetic operations involving ring elements are performed in $\mathcal{Z}_p$. The shares of the parties are specified in Table 3.1. We turn to prove that this secret-sharing scheme satisfies the correctness and privacy properties with respect to $\mathbf{NQRP}_p$. Let SUM_w denote the sum of the m shares held by parties in B_w. Both the correctness and the privacy proofs will rely on the following lemma.

LEMMA 3.3. $\mathrm{SUM}_w = w^s r^2$.

Proof. By (3.1) we obtain the following:

- If $s=0$, then
$$\mathrm{SUM}_w = \sum_{i=0}^{m-1} z_i + r^2 = r^2.$$
- If $s=1$, then
$$\begin{aligned}\mathrm{SUM}_w &= \sum_{i=0}^{m-1} (z_i + w_i 2^i r^2) \\ &= \sum_{i=0}^{m-1} z_i + r^2 \sum_{i=0}^{m-1} (w_i 2^i) \\ &= r^2 w. \qquad \square\end{aligned}$$

Correctness. We separately consider two types of minimal authorized sets B.

- $B = \{P_i^0, P_i^1\}$ for some $0 \le i < m$. In this case, $s=0$ iff the shares of P_i^0 and P_i^1 are equal. This follows from the fact that $2^i r^2 \not\equiv 0 \bmod p$ for every i.
- $B = B_w$ for some w such that w is not a quadratic residue. In this case, it follows from Lemma 3.3 that $s=0$ iff SUM_w is a quadratic residue (since the product of a quadratic residue and a nonquadratic residue is a nonquadratic residue).

[9] To make the access structure ZPP-uniform, p can be chosen to be the least prime in the interval $[2^{\lceil n/2 \rceil}, 2^{\lceil n/2 \rceil} + n]$, or 3 if none exists. However, as for other number-theoretic functions, a random choice of p may be safer when assuming that **NQRP** is not in NC.

Privacy. We need to prove that every unauthorized set $C \notin \mathbf{NQRP}_p$ has no information on the secret. It suffices to prove this claim for every *maximal* C not in the access structure. There are two cases to consider.

- $C = B_w$ for some $w \in \{0,1\}^m$ such that w is a quadratic residue. In this case we claim that, regardless of the value of the secret, the share-vector of the parties in C is uniformly distributed over the m-tuples of field elements whose sum is a quadratic residue. Indeed, by Lemma 3.3, if $s = 0$, then $\mathrm{SUM}_w = r^2$, which is a uniformly random quadratic residue. Furthermore, fixing the choice of r, the choices of z_i induce a uniformly random share-vector among all those which sum to r^2. Similarly, if $s = 1$, then $\mathrm{SUM}_w = r^2 w$. Since w is a quadratic residue, SUM_w is again a uniformly random quadratic residue determined by r, and the same argument as above applies.
- $C = B_w \setminus \{P_j^{w_j}\}$ for some $w \in \{0,1\}^m$ and $0 \leq j < m$. That is, C is a set of size $m-1$ such that for exactly one j it contains neither P_j^0 nor P_j^1. We claim that in this case the share-vector of the parties in C is uniformly distributed in $\mathcal{Z}_p^{m-1}$, regardless of the secret. It suffices to show that for every secret $s \in \{0,1\}$, every possible value of the share-vector from $\mathcal{Z}_p^{m-1}$, and every fixed $r_0 \in \mathcal{Z}_p^*$, there exists a unique choice of $z_0, \ldots, z_{m-2}$ generating this value with $r = r_0$. This can be verified by inspection of the corresponding system of linear equations over $\mathcal{Z}_p$. $\square$

A generalization of our construction for **NQRP** is described in Appendix A. This generalization will uncover what algebraic properties we use in our construction, and will supply us with a few more examples.

3.1. Does NQRP have an efficient linear secret-sharing scheme? The access structure **NQRP** we have realized above is related to the problem of deciding quadratic residuosity modulo a prime. We would like to argue that **NQRP** is likely not to be in NC, which would imply in particular that **NQRP** cannot be efficiently realized by linear schemes. We start by describing some known facts about the complexity of the quadratic residuosity problem.

Unlike quadratic residuosity modulo a composite, whose intractability is commonly assumed in cryptography (see [39]), quadratic residuosity modulo a prime can be decided in polynomial time. All known algorithms for this problem are sequential. It is not known if efficient parallel algorithms for this problem exist; that is, the situation is similar to the exponentiation function and the gcd function. There are two types of known algorithms. The first uses Euler's criterion, which states that w is a quadratic residue modulo an odd prime p iff $w^{(p-1)/2} \equiv 1 \bmod p$. Thus, this algorithm requires modular exponentiation. For a survey of algorithms for exponentiation, see [40]. The second type of algorithm computes the Jacobi symbol in a way similar to Euclid's algorithm for computing the gcd. For more details, see, e.g., [3, Chapter 5]. "Weak" parallel algorithms for checking quadratic residuosity follow from the algorithms of [34] for computing the Jacobi symbol and the algorithm of [1] for exponentiation. More precisely, there is (1) an algorithm that runs in $O(n/\log\log n)$ time using $O(n^{1+\epsilon})$ processors [34], (2) an algorithm that runs in $O(\log^2 n \log\log n)$ time using $2^{O(n/\log n)}$ processors [34], and (3) an algorithm that runs in $O(\log^3 n)$ time using $2^{O(\sqrt{n \log n})}$ processors [1].

The best known polynomial-size circuit for the quadratic residuosity problem has depth $O(n/\log\log n)$ where $n = \log p$ [34]. Thus, given the current state of knowledge on this problem and the related modular exponentiation problem, it is reasonable to assume that they are not in NC. In fact, this assumption (for the exponentiation problem) has been explicitly relied on in [16].

It is easy to see that deciding quadratic residuosity modulo p can be very efficiently reduced to computing the monotone function defined by $\mathbf{NQRP}_p$. However, there is a major difference between the "standard" algorithmic setting for this problem and our setting. Our setting is highly nonuniform, in the sense that with each input length (or number of parties) we associate some *fixed* prime p. Hence, when computing this access structure one may use a nonuniform polynomial-size "advice" depending on p. In algorithmic terms, we allow unlimited preprocessing which depends on the prime p but not on the other input w. Nevertheless, we do not know how to use this type of preprocessing to obtain an efficient parallel algorithm for the quadratic residuosity problem.[10] (It is interesting to note, however, that deciding quadratic residuosity modulo a *composite* is no more difficult in our setting than deciding quadratic residuosity modulo a prime, since the factorization of the composite may be used as advice.) To conclude, the assumption that $\mathbf{NQRP} \notin \mathrm{NC}$ is stronger than the assumption that the standard quadratic residuosity problem (or modular exponentiation) is not in NC, although this still seems very reasonable given the current state of knowledge.

In light of the uncertain situation described above, one could hope for an unconditional superpolynomial lower bound on a size of a *monotone span program* computing **NQRP**. This would be sufficient for proving that **NQRP** cannot be efficiently realized by linear schemes and, as noted in the introduction, there are explicit monotone functions for which such bounds are known. However, as we argue next, such lower bounds are impossible to prove for the **NQRP** structure, as well as for the access structures considered in section 4, without proving that $\mathrm{NC}^1 \neq \mathrm{P}$. For a fixed $(m+1)$-bit prime p, the quadratic residuosity function (modulo p) is defined as $f_p(x_0, \ldots, x_{m-1}) = 1$ iff $\sum_{i=0}^{m-1} x_i 2^i$ is a quadratic residue modulo p. This function is not monotone. To define the monotone access structure **NQRP** we replaced each literal by two parties, obtaining an access structure with $2m$ parties. (This is a standard transformation, e.g., when proving that monotone circuit evaluation is P-complete [38].) For technical reasons we also added m minterms of size two. It follows that the monotone formula size of $\mathbf{NQRP}_p$ is *equal*, up to an additive $O(n)$ difference, to the (nonmonotone) formula size of the function f_p. Thus, one cannot expect to prove superpolynomial lower bounds on the size of a monotone span program (or even a monotone formula) for **NQRP**, since they will imply, in particular, superpolynomial lower bounds on the (nonmonotone) formula size of the quadratic residuosity function.[11]

4. An efficient nonlinear scheme: The statistical case.

In this section we construct an efficient nonlinear secret-sharing scheme whose access structure is as hard as the general quadratic residuosity function. Unlike the previous construction, the scheme we construct below is only statistically private and correct, and its reconstruction procedure is computationally inefficient. In section 4.1 we show that perfect correctness (but not perfect privacy) can be achieved under a number-theoretic assumption, namely, the extended Riemann hypothesis. We end this section by discussing a generalization of our construction which applies to the so-called t-residuosity problem. As a special case, we obtain an efficient scheme whose access structure is computationally equivalent to the coprimality problem.

[10] Preprocessing can parallelize the algorithms for exponentiation when the field size and the exponentiation base are given in advance (see [40]). However, in our case we know in advance the field size and the exponentiation power.

[11] The best known lower bound on the formula size for an explicit function is $\Omega(n^{3-o(1)})$ [41].

Table 4.1
A secret-sharing scheme for $\mathbf{NQR}_m$.

	$s=0$	$s=1$
W_0^b, where $b \in \{0,1\}$	$r^2 + z_0$	$br^2 + z_0$
W_i^b, where $1 \le i < m,\ b \in \{0,1\}$	z_i	$2^i br^2 + z_i$
U_i^b, where $0 \le i < m,\ b \in \{0,1\}$	$2^i br' + z_{i+m}$	$2^i br' + z_{i+m}$

Definition 4.1 (the nonquadratic residue access structure ($\mathbf{NQR}_m$)). *Let m be a positive integer. We define the n-party access structure $\mathbf{NQR}_m$, where $n \overset{def}{=} 4m$, by specifying its collection of minimal sets. It will be convenient in what follows to denote the first $2m$ parties by W_i^b and the last $2m$ parties by U_i^b, where $b \in \{0,1\}$ and $0 \le i < m$. With each pair (w,u), where $w,u \in \{0,1\}^m$, we naturally associate a subset of parties $B_{w,u}$ of size $2m$, defined by*

$$B_{w,u} \overset{def}{=} \{W_i^{w_i} : 0 \le i < m\} \cup \{U_i^{u_i} : 0 \le i < m\}.$$

We will freely identify strings w,u as above with integers in the interval $[0, 2^m - 1]$. A set B is a minimal set of $\mathbf{NQR}_m$ *if*

1. *$B = \{W_i^0, W_i^1\}$ or $B = \{U_i^0, U_i^1\}$ for some $0 \le i < m$, or*
2. *$B = B_{w,u}$ for some w,u such that w is* not *a quadratic residue modulo u. (For technical reasons, we assume here that this condition never holds when $u = 1$ and always holds when $u = 0$ except when $w = 1$.)*

We let $\mathbf{NQR}$ *denote the family of access structures in which the nth structure is* $\mathbf{NQR}_{\lfloor n/4 \rfloor}$.

We start by observing that the computational complexity of the access structure $\mathbf{NQR}$ is essentially the same as that of the general quadratic residuosity problem.

Claim 4.2. *The circuit complexity of* $\mathbf{NQR}$ *is the same, up to an $O(n)$ difference, as that of the Language*

$$\{(w,u) : |w| = |u| \text{ and } w \text{ is a quadratic residue modulo } u\}.$$

It follows that, under the quadratic residuosity assumption [39], computing $\mathbf{NQR}$ requires circuits of superpolynomial size. The remainder of this section will be devoted to proving the existence of an efficient nonlinear secret-sharing scheme for $\mathbf{NQR}$. Specifically, we show the following theorem.

Theorem 4.3. *There exists a statistical secret-sharing scheme for* $\mathbf{NQR}_m$ *in which*

- *the secret-domain is $\{0,1\}$,*
- *the share size of each party is $O(k^2 + km)$ (where k is the security parameter),*
- *the reconstruction error probability is 2^{-k},*
- *the privacy level is $\epsilon(k) = O(k/2^k)$.*

Our secret-sharing scheme for $\mathbf{NQR}_m$ proceeds as follows. Let $D \overset{\text{def}}{=} 2^{4m+3k+1}$. In what follows, all arithmetic operations will be performed in $\mathcal{Z}_D$. The dealer chooses $z_0, z_1, \ldots, z_{2m-1} \in \mathcal{Z}_D$ at random subject to the restriction that they sum to 0. In addition, it chooses two random integers $1 \le r \le 2^{m+k}$ and $1 \le r' \le 2^{3(m+k)}$. Each party receives a single element of $\mathcal{Z}_D$, as specified in Table 4.1. For amplifying the correctness probability, the above distribution procedure should be independently repeated k times, so that each party receives k elements of $\mathcal{Z}_D$. In addition, the minimal authorized sets of size 2 should be taken care of separately, by independently

```
Let $\mathrm{SUM}_{w,u}$ be the sum of the $2m$ shares held by parties in $B_{w,u}$.
If $\gcd(w,u) = 1$, then (* $w$ is a quadratic nonresidue modulo $u$ *).
    If $\mathrm{SUM}_{w,u}$ is a quadratic residue modulo $u$, then $s = 0$ else $s = 1$.
If $\gcd(w,u) \neq 1$, then
    Let $c = \gcd(w,u)$,
    If $c$ divides $\mathrm{SUM}_{w,u}$, then $s = 1$ else $s = 0$.
```

FIG. 4.1. *Reconstruction procedure for $B_{w,u}$ in $\mathbf{NQR}_m$.*

sharing s among each such authorized pair (that is, for each such pair choose an independent random bit α, and give α to the first party and $\alpha \oplus s$ to the second).[12] This adds only a single bit to the size of each share. The following analysis will mostly focus on the core of the scheme, as described in Table 4.1.

Statistical correctness. The minimal authorized sets of size 2 were explicitly taken care of in the above construction. It thus remains to prove the correctness for a subset $B_{w,u}$, where w is not a quadratic residue modulo u. The following lemma, which can be verified by inspection of Table 4.1, is used to show how to reconstruct the secret.

LEMMA 4.4. *Let* $\mathrm{SUM}_{w,u}$ *be the sum of the* $2m$ *shares held by parties in* $B_{w,u}$. *Then,* $\mathrm{SUM}_{w,u} = r^2 w^s + r'u$ *for any* $0 \leq w, u < 2^m$ *and secret* $s \in \{0,1\}$.

Note that by our choice of parameters, the expression $r^2 w^s + r'u$ in Lemma 4.4 is always less than D. We will therefore treat this expression as being evaluated over the integers.

If r was chosen such that $\gcd(r,u) = 1$, then the correctness would follow from similar arguments to those of the proof for **NQRP** (that is, the secret is reconstructed by checking if the sum of shares is a quadratic residue modulo u). However, since here u is not fixed, we cannot guarantee that the above condition always holds. Nevertheless, this already implies that the secret can be correctly reconstructed from the shares in Table 4.1 with a one-sided error probability of at most $1 - \varphi(u)/u$, which is bounded away from 1 (that is, it is $1 - O(1/\log\log u)$). The following tighter analysis, which does not assume that $\gcd(r,u) = 1$, shows that the one-sided error probability of reconstruction is at most $1/2$. Hence, with k independent repetitions the error probability is at most 2^{-k}. In Figure 4.1 we present the reconstruction procedure and then prove its correctness.

From now on, we assume that $u \geq 2$ (the case that $u = 0$ and $w \neq 1$ can be verified separately). Suppose first that $\gcd(w,u) = c > 1$, and let $c' > 1$ be a prime dividing c. In this case, c always divides $r^2 w + r'u$, whereas c divides $r^2 + r'u$ implies that c' divides r. Thus, with probability at least $1 - 1/c' \geq 1/2$, the gcd c does not divide $r^2 + r'u$. It follows that when $\gcd(w,u) > 1$ the case $s = 0$ can be distinguished from the case $s = 1$ with a one-sided error probability of at most $1/2$, as described above.

Now suppose that w is a quadratic nonresidue modulo u. In this case, $r^2 + r'u \equiv r^2 \bmod u$ is always a square modulo u. This implies that $\mathrm{SUM}_{w,u}$ is a quadratic residue when $s = 0$. The following lemma shows that with probability at least $1/2$, this is not the case for $r^2 w + r'u$, i.e., when $s = 1$.

LEMMA 4.5. *Suppose that* w *is a quadratic nonresidue modulo* u *(in particular,*

[12] In fact, as in the previous construction this additional sharing is unnecessary for sets of the form $\{W_i^0, W_i^1\}$.

$\gcd(w,u)=1$*). Then, the probability that* r^2w *is a quadratic residue modulo* u *is at most* $1/2$.

Proof. By the Chinese remainder theorem, a number is a quadratic-residue modulo u iff it is a quadratic-residue modulo each prime power dividing u. Thus, there exists a prime power p^α dividing u such that w is a quadratic nonresidue modulo p^α. Now, if wr^2 is a square modulo u, then it is also a square modulo p^α, and so there exists d such that $d^2 \equiv wr^2 \bmod p^\alpha$. We argue that it must be the case that p divides r. Otherwise, r has an inverse modulo p^α and $w \equiv (d/r)^2 \bmod p^\alpha$, contradicting the fact that w is a quadratic nonresidue modulo p^α. The lemma follows by noting that the probability that p divides r is at most $1/p \le 1/2$, as required. □

This concludes the analysis of the reconstruction procedure described above. Note that this reconstruction procedure is computationally inefficient if the factorization of u is unknown.

Statistical privacy. We now prove the privacy of our construction. As before, it suffices to consider maximal unauthorized sets of two types. The first type consists of sets C such that $|C| < 2m$ and C does not contain a pair W_i^0, W_i^1 or a pair U_i^0, U_i^1. For such a set C, it can be verified that the shares received by its parties are uniformly and independently distributed over $\mathcal{Z}_D$, regardless of the secret s.

We turn to the more interesting case of a set $C = B_{w,u}$ such that $u \ge 2$ and w is a quadratic residue modulo u. (The cases $u=1$ and $u=0, w=1$ can be verified separately.) When $s=0$ the shares are random subject to the restriction that their sum is $r^2 + r'u$, and when $s=1$ the shares are random subject to the restriction that their sum is $r^2w + r'u$. Thus, it suffices to show that in this case $\mathrm{SD}(r^2+r'u, r^2w+r'u) = O(2^{-k})$. We prove this using the following lemmas. In the lemmas we denote by r and r' the random variables used in the scheme (taking uniform integral values from the intervals $[1, 2^{m+k}]$ and $[1, 2^{3(m+k)}]$, respectively). For the proof we also use an additional random variable r_u which is a uniformly distributed integer in $[0, u-1]$.

LEMMA 4.6. *If* w *is a quadratic residue modulo* u*, then the distribution of* $(wr_u^2) \bmod u$ *is identical to that of* $r_u^2 \bmod u$.

Proof. Since w is a quadratic residue modulo u, there exists b such that $\gcd(b,u) = 1$ and $b^2 \equiv w \bmod u$. Since $wr_u^2 \equiv (br_u)^2 \bmod u$, it suffices to show that $(br_u) \bmod u$ is identically distributed to $r_u \bmod u = r_u$. Finally, since $\gcd(b,u)=1$, i.e., b has an inverse modulo u, then $\Pr[br_u \equiv \beta] = \Pr[r_u \equiv (\beta/b)] = 1/u$ for every value β. □

LEMMA 4.7. $\mathrm{SD}(r^2 \bmod u\,,\, r_u^2 \bmod u) \le 2^{-k}$.

Proof. Recall that r is chosen uniformly from the interval $[1, 2^{m+k}]$. If u divides 2^{m+k}, then the above two distributions are identical. Otherwise, the contribution of each $y \in [0, u-1]$ to this distance is at most $1/2^{m+k}$, and since $u < 2^m$ the total contribution is at most $2^m/2^{m+k} = 2^{-k}$. □

From the previous two lemmas, we may conclude that

$$\mathrm{SD}(wr^2 \bmod u, r^2 \bmod u) = O(2^{-k}). \tag{4.1}$$

Now, define the multisets

$$V = \left\{wr^2 \bmod u \,:\, 1 \le r \le 2^{m+k}\right\}$$

and

$$Z = \left\{r^2 \,:\, 1 \le r \le 2^{m+k}\right\}.$$

Let Z' be a maximal multiset such that $Z' \subseteq Z$ and $Z' \bmod u \overset{\text{def}}{=} \{z \bmod u \,:\, z \in Z'\} \subseteq V$. It follows from (4.1) that $|Z'| = (1 - O(2^{-k}))|Z|$. Define $S = Z' \cup (V \setminus (Z' \bmod u))$. Note that $|S| = |V| = 2^{m+k}$. We will denote the elements of S by $y_1, \ldots, y_{2^{m+k}}$ and the uniform distribution over S by Y. It follows from the above information that Y satisfies the following: (1) $\mathrm{SD}(Y, r^2) = O(2^{-k})$; (2) the distribution of $Y \bmod u$ is *identical* to that of $wr^2 \bmod u$; and (3) $Y \leq 2^{2(m+k)}$.

We would like to conclude that $\mathrm{SD}(wr^2 + r'u, r^2 + r'u) = O(2^{-k})$. To this end, we use the following lemma.

Lemma 4.8. *Let y, z be two integers in some interval $[0, M]$ such that $y \equiv z \bmod u$, and let R be uniformly distributed in the interval $[1, MK]$. Then,* $\mathrm{SD}(y + Ru\,,\, z + Ru) \leq 1/K$.

Proof. The statistical distance is bounded by $|y - z|/(uMK) \leq M/(uMK) < 1/K$. □

We are now ready to complete the proof of privacy. From Property (1) of Y it follows that

$$\mathrm{SD}(Y + r'u, r^2 + r'u) = O(2^{-k}). \tag{4.2}$$

From Property (2) of Y, we may assume that $y_r \equiv wr^2 \bmod u$ for every $1 \leq r \leq 2^{m+k}$. Letting $M = 2^{3m+2k}$ and $K = 2^k$, both Y and wr^2 are no larger than M, and r' is uniform in $[1, MK]$. Since

$$\mathrm{SD}(Y + r'u, wr^2 + r'u) \leq E_r[\mathrm{SD}(y_r + r'u, wr^2 + r'u)]$$

it follows from Lemma 4.8 that

$$\mathrm{SD}(Y + r'u, wr^2 + r'u) \leq 2^{-k}. \tag{4.3}$$

Combining (4.2) and (4.3) we get that $\mathrm{SD}(wr^2 + r'u, r^2 + r'u) = O(2^{-k})$, as required.

As explained above, to reduce the error probability in the reconstruction from $1/2$ to 2^{-k} we share the secret independently k times. By standard arguments, this can only increase the statistical distance to $O(k/2^k)$, which is still negligible in k.

4.1. A perfectly correct scheme. In this section we show that under the extended Riemann hypothesis (ERH), one can obtain a variant of the above scheme which is *perfectly* correct, though still only statistically private. (It is open if there is a scheme with perfect correctness and privacy which efficiently realizes **NQR**.) The only required modification is the choice of r: instead of choosing it uniformly from the interval $[1, 2^{m+k}]$, it is chosen as a random *prime* from the interval $[2^m, 2^{m+k}]$. Since $u < 2^m$, this guarantees that r is relatively prime to u, and this in turn is sufficient to guarantee perfect correctness. We next argue that under the ERH, the resulting scheme is statistically private.

We will need the following results on the distribution of primes. For more information on this subject the reader might consult, e.g., [3, Chapter 8]. For an integer x let $\pi(x)$ be the number of primes in the interval $[1, x]$, and for integers x, w, and u let $\pi(x, u, w)$ be the number of primes in the interval $[1, x]$ that are congruent to $w \bmod u$. It is known that $\pi(x) \approx x/\log x$. If $\gcd(w, u) > 1$, then every number that is congruent to $w \bmod u$ is a composite. It turns out that the primes are nearly uniformly distributed among the other residue classes modulo u. That is, if $\gcd(w, u) = 1$, then $\pi(x, u, w) \approx \frac{1}{\varphi(u)} x/\log x$, where $\varphi(u)$ is the Euler function of u.

We will need good bounds on the error terms in the above approximations. The bounds that can be proved unconditionally are too crude for our purpose, and we will need bounds based on the ERH. Proving this famous hypothesis is one of the most important open questions in mathematics. We will not formulate the statement of this hypothesis, and only state the following conclusion from the ERH. The estimations that are used to derive the next theorem are presented in Appendix B, where it is also shown how to derive Theorem 4.9 from these estimations.

THEOREM 4.9. *If the ERH holds and* $\gcd(w,u) = 1$*, then for every* x *and* x'*, where* $u \le x' \le \sqrt{x}$*,*

$$\left| \frac{\pi(x,u,w) - \pi(x',u,w)}{\pi(x) - \pi(x')} - \frac{1}{\varphi(u)} \right| = O\left(\frac{\log^2 x}{\sqrt{x}}\right),$$

where the constant in the "O" notation is an absolute constant independent of w*,* u*, and* x*.*

Notice that $\frac{\pi(x,u,w)-\pi(x',u,w)}{\pi(x)-\pi(x')}$ is the probability that a uniformly random prime in the interval $[x', x]$ is congruent to u modulo w. Thus, the above theorem states that this probability is close to the probability that a uniformly random element from $\mathcal{Z}_u^*$ is equal to w.

COROLLARY 4.10. *Let* $u < 2^m$*,* U *be a random variable distributed uniformly in* $\mathcal{Z}_u^*$*, and* r *be a uniformly chosen prime in the interval* $[2^m, 2^{m+k}]$*. If the ERH holds, then* $\mathrm{SD}(U, r \bmod u) \le 2^{-\Omega(k)}$ *for every* k *and* m *such that* $k \ge 3m$*, and in particular* $\mathrm{SD}(U^2 \bmod u, r^2 \bmod u) \le 2^{-\Omega(k)}$*.*

Proof.

$$\begin{aligned}
\mathrm{SD}(U, r \bmod u) &= \frac{1}{2} \sum_{y \in \mathcal{Z}_u^*} |\Pr[U = y] - \Pr[r \bmod u = y]| \\
&\le \varphi(u) \cdot O\left(\frac{(m+k)^2}{\sqrt{2^{m+k}}}\right) \\
&= O\left(\frac{2^m (m+k)^2}{2^{0.5(m+k)}}\right) = 2^{-\Omega(k)}.
\end{aligned}$$

The last equality holds since $k \ge 3m$. □

To guarantee that the statistical distance decreases exponentially with the security parameter *independently of* m, we execute the scheme with $k' = \max(k, 3m)$. Closely following the privacy proof of the previous protocol (and replacing Lemma 4.7 with Corollary 4.10), one can show that the scheme is statistically private with $\epsilon(k) = 2^{\Omega(-k)}$. The next theorem summarizes the properties of this scheme.

THEOREM 4.11. *If the ERH holds, then there exists a statistical secret-sharing scheme for* $\mathbf{NQR}_m$ *with perfect correctness in which*

- *the secret-domain is* $\{0, 1\}$*,*
- *the share size of each party is* $O(k + m)$*,*
- *the privacy level is* $\epsilon(k) = 2^{-\Omega(k)}$*.*

4.2. Schemes for *t*-residuosity. The quadratic residuosity problem naturally generalizes to the t-residuosity problem defined as follows. An integer w is a t-residue modulo u if $\gcd(w,u) = 1$ and there exists an integer b such that $w \equiv b^t \bmod u$. The non-t residue access structure (**N*t*R**) is defined as the access structure **NQR**, with quadratic residuosity replaced by tth residuosity.

A scheme for **N*t*R** can be obtained by the following small modification to the scheme for **NQR**: the ring size D is changed to $2^{(t+2)m+(t+1)k+1}$, the random string r' is chosen with uniform distribution from $[1, 2^{(t+1)(t+k)}]$, and in the dealer's distribution procedure we replace r^2 by r^t. The correctness and privacy of the modified scheme are argued similarly to the original scheme. These modification also work in the scheme based on the ERH.

An interesting special case of the general scheme is when $t = 1$. In the resultant scheme, $B_{w,u}$ can reconstruct the secret iff the integers w and u are not coprimes (i.e., $\gcd(w, u) > 1$). Hence, its access structure is computationally equivalent to the coprimality problem. Checking if two integers are coprimes is clearly in P, and it is not known to be in NC. The best parallel algorithms for the coprimality problem compute the gcd. The question if the gcd can be computed in parallel, that is, with polylogarithmic time and polynomial number of processors, was first raised by Cook [25] and is still open. Parallel algorithms with sublinear time, namely, $O(n/\log n)$ time, and polynomial number of processors were presented by [44, 24, 56]. Parallel algorithms with polylogarithmic time and subexponential number of processors were presented by [1]. An important feature of this instance of the general construction is that it is computationally efficient: indeed, reconstruction only requires checking if $\gcd(w, u)$ divides $\mathrm{SUM}_{w,u}$.

5. Quasi-linear secret-sharing.

In this section we study a natural extension of the class of linear secret-sharing schemes to what we call *quasi-linear* schemes. Quasi-linear schemes are obtained by *composing* a finite number of linear secret-sharing schemes, possibly over different fields.

Towards defining quasi-linear schemes, it will be convenient to use the following notation for extending the secret-domain of a given secret-sharing scheme to an arbitrarily large finite domain.

DEFINITION 5.1. *Let Π be a secret-sharing scheme with secret-domain S and share-domains $S_0, \ldots, S_{n-1}$, let $T = \{0, 1, \ldots, |T| - 1\}$ be any finite secret-domain, and let $\ell = \lceil \log_{|S|} |T| \rceil$. Then, by $\tilde{\Pi}_T$ we denote the randomized mapping from T to $S_0^\ell \times \cdots \times S_{n-1}^\ell$ defined as follows. For a secret $t \in T$, let $(t_1, \ldots, t_\ell)$ denote its base-$|S|$ representation, where $t_i \in S$ for all i. The output of $\tilde{\Pi}_T(t)$ is obtained by independently applying Π to each t_i and letting the ith entry of the output be the concatenation of the ith entries from the ℓ outputs of Π.*

As can be easily seen, $\tilde{\Pi}_T$ defines a secret-sharing scheme realizing the same access structure as Π, whose secret-domain is T and whose share-complexity is $\ell = \lceil \log_{|S|} |T| \rceil$ times that of Π. We are now ready to formally define the notion of quasi-linear schemes.

DEFINITION 5.2 (quasi-linear secret-sharing). *An n-party quasi-linear secret-sharing scheme is recursively defined as follows:*

1. *Any n-party linear secret-sharing scheme is an n-party quasi-linear scheme.*
2. *Suppose that Π is an n'-party linear scheme over a field F with share-domains $S_0, \ldots, S_{n'-1}$, and let $\Pi^0, \ldots, \Pi^{n'-1}$ be n-party quasi-linear schemes. Then, define an n-party quasi-linear secret-sharing scheme $\Pi(\Pi^0, \ldots, \Pi^{n'-1})$ with secret-domain F as follows. To share $s \in F$, first apply $\Pi(s)$ to obtain shares $s_0, \ldots, s_{n'-1}$. Then, identifying each share-domain S_i with the set $\{0, 1, \ldots, |S_i| - 1\}$, independently share each s_i among the n parties using $\tilde{\Pi}^i_{S_i}$.*

It is convenient to view an n-party quasi-linear scheme Π as a tree, in which every node contains a linear secret-sharing scheme. Associating each linear scheme with its corresponding monotone span program, we may view this tree as a Boolean formula

φ_Π over the basis of all monotone span programs (over all finite fields);[13] that is, each gate in the formula computes the Boolean function computed by a monotone span program. For brevity we refer to such a formula as an *MSP-formula*.

The following proposition establishes the correspondence between a quasi-linear scheme and its associated MSP-formula. Its proof is a generalizing of the proof for the AND-OR-threshold formula construction from [10].

PROPOSITION 5.3. *Let* Π *be a quasi-linear secret-sharing scheme and* φ_Π *be the corresponding MSP-formula. Then,* Π *realizes the access structure computed by* φ_Π.

The scheme $\Pi(\Pi^0, \ldots, \Pi^{n'-1})$ from case (2) in Definition 5.2 is just the standard definition of composition of Π with $\Pi^0, \ldots, \Pi^{n'-1}$, thus, a formal proof of Proposition 5.3 follows, by induction, from, e.g., [47, 48].

Beimel and Weinreb [8] proved that quasi-linear schemes are strictly stronger than linear schemes. More precisely, they proved that there are explicit functions that have small quasi-linear schemes; however, they require linear schemes of size $n^{\Omega(\log n)}$. We show next that quasi-linear schemes cannot be too powerful. More specifically, if there is an efficient quasi-linear scheme for f, then f can be computed by a shallow circuit. The idea of the proof is to consider the corresponding MSP-formula φ. We use a result of [5] showing that a formula φ over a general basis can be "balanced" to obtain an equivalent formula whose depth is small and its size is not too big (this is a generalization of the well-known result from [57] for bounded fan-in formulae over the standard basis). An instantiation of this result, which is useful for our purposes, is quoted in the following lemma.

LEMMA 5.4 (see Beigel and Fu [5]). *Let* φ *be a MSP-formula. Then, there exists a MSP-formula* $\hat{\varphi}$ *such that* (1) $\hat{\varphi}$ *computes the same function as* φ, (2) *the depth of* $\hat{\varphi}$ *is* $O(\log(\text{size}(\varphi)))$, (3) *the size of* $\hat{\varphi}$ *is* $\text{size}(\varphi)^{O(1)}$, *and* (4) *each node of* $\hat{\varphi}$ *is either labeled by some span program appearing in* φ, *or is labeled by an AND, OR, or NOT gate.*

THEOREM 5.5. *Suppose that* f *is efficiently realized by quasi-linear schemes. Then,* $f \in \mathrm{NC}^4$.

Proof. Let Π be an efficient quasi-linear scheme realizing f, and let φ be the corresponding MSP-formula. We may assume without loss of generality that the span program labeling each *internal* node of φ depends on all of its inputs, and has at least two inputs; otherwise, Π could be simplified into a quasi-linear scheme Π' whose MSP-formula φ' satisfies this property. As the number of leaves in such a φ is a lower bound on the complexity of Π (and the degree of each internal node of φ is at least 2), φ must be of size $\text{poly}(n)$. It also follows that each node v of φ must be labeled by a *polynomial-size* monotone span program M_v over a field $\mathrm{GF}(q_v)$ such that $\log q_v = \text{poly}(n)$.[14] By Lemma 2.7, the function f_v computed by M_v can be simulated by a Boolean circuit of size $\text{poly}(n)$ and depth $O(\log^3 n)$. The theorem follows by applying Lemma 5.4 to φ and replacing each node in $\hat{\varphi}$ by a corresponding NC^3 circuit. □

We conclude this section by showing an application of quasi-linear schemes for the construction of secret-sharing schemes efficiently realizing monotone span programs over a ring $\mathcal{Z}_u$, where u is a square-free composite.[15]

[13] An input variable is viewed as a size-1 monotone span program in the variables $x_0, \ldots, x_{n-1}$ returning its value.

[14] The converse does not hold. It is easy to construct a polynomial-size MSP-formula (even a shallow one) which is efficient in this sense, but whose corresponding quasi-linear scheme is inefficient.

[15] Span programs over rings are defined in a completely analogous way to span programs over fields.

THEOREM 5.6. *Let $\widehat{M} = \langle M, \rho, \vec{v} \rangle$ be a monotone span program over $\mathcal{Z}_u$, where u is the product of k distinct primes $p_1, \ldots, p_k$. Then, there exists a quasi-linear scheme Π_M realizing the access structure defined by M, whose share-complexity is* $\mathrm{size}(M) \cdot \sum_{j=1}^{k} \lceil \log p_j \rceil = O(\mathrm{size}(M) \cdot \log u)$.

Proof. The scheme Π_M is defined by the following depth-2 MSP-formula φ_M. The root contains an AND gate with fan-in k (represented by a size-k monotone span program over GF(2)). The jth leave, $1 \leq j \leq k$, contains a monotone span program $\widehat{M_j} = \langle M_j, \rho, \vec{v}_j \rangle$ over GF(p_j), obtained from $\widehat{M}$ by reducing each of M and $\vec{v}$ entries modulo p_j. By Proposition 5.3, to prove that Π_M indeed realizes the access structure defined by $\widehat{M}$ it suffices to show that φ_M computes the same function as $\widehat{M}$. Indeed, if $M(x) = 1$, then clearly $M_j(x) = 1$ for all j (as witnessed by the same linear combination, modulo p_j). The converse follows by applying the Chinese remainder theorem to the k linear combination vectors witnessing that $M_j(x) = 1$, where $1 \leq j \leq k$.[16] □

Example 5.7. Figure 5.1 shows an efficient span program over $\mathcal{Z}_u$ for testing whether the input x (viewed as an integer) is coprime to u. Replacing each negative literal with a new variable, we get a *monotone* span program for an access structure whose complexity is equivalent to deciding whether x is coprime to some *fixed* integer u.[17] Using Theorem 5.6, we get a very efficient quasi-linear scheme for this access structure. We note that the scheme from section 4.2 is stronger in the sense that it efficiently applies to the standard coprimality problem (with no fixed inputs). However, this scheme only realizes the relaxed notion of statistical secret-sharing.

$$\begin{array}{r|cccccccc}
\bar{x}_0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\
x_0 & 1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\
\bar{x}_1 & 0 & -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\
x_1 & 2 & -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & & & & \ddots & & & \vdots \\
\bar{x}_{n-2} & 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 1 \\
x_{n-2} & 2^{n-2} & 0 & 0 & 0 & \cdots & 0 & -1 & 1 \\
\bar{x}_{n-1} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & -1 \\
x_{n-1} & 2^{n-1} & 0 & 0 & 0 & \cdots & 0 & 0 & -1 \\
\hline
\text{target} & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0
\end{array}$$

FIG. 5.1. *A span program over $\mathcal{Z}_u$ testing whether* $\gcd(x, u) = 1$.

Appendix A. A generalization of the scheme from section 3. In this section we show how to generalize the scheme for **NQRP** to similar access structures. This generalization will uncover what algebraic properties we use in our construction, and will supply us with a few more examples.

[16] If $\vec{v}_j = \vec{0}$ for some j, then $\widehat{M}_j$ should accept every input (as witnessed by the trivial combination of rows). However, in the definition of span programs we require that the target vector is a nonzero vector. Thus, Π_M has a leaf for every j such that $\vec{v}_j \neq \vec{0}$.

[17] Whether x is coprime to u can be tested in NC^1 given advice depending on u (namely, its factorization). Hence, there exist efficient linear secret-sharing schemes for this access structure. Still, the exact efficiency of the quasi-linear scheme is much better. See Example A.2 for an efficient nonlinear realization which does not rely on the factorization of u.

Let $R = \langle A, +, *\rangle$ be a finite ring and $B \subseteq A \setminus \{0\}$ be such that $G = \langle B, *\rangle$ is a group.[18] In the sequence, all arithmetic operations involving ring elements are performed in the ring. We assume that $2 * a \neq 0$ for every $a \in R \setminus \{0\}$. We define the access structure $\mathcal{A}_{R,G}$ in a similar way to **NQRP**.

Definition A.1 (the access structure $\mathcal{A}_{R,G}$). *Let $m \overset{def}{=} \lfloor \log |R| \rfloor$. We define the n-party access structure $\mathcal{A}_{R,G}$, where $n \overset{def}{=} 2m$, by specifying its collection of minimal sets. With an integer $w \in \{0,1\}^m$ we naturally associate a set B_w of size m defined by*

$$B_w \overset{def}{=} \{P_i^{w_i} : 0 \leq i < m\}.$$

A set B is a minimal set of $\mathcal{A}_{R,G}$ if

- *$B = \left\{P_i^0, P_i^1\right\}$ for some $0 \leq i < m$, or*
- *$B = B_w$ for some $w \in \{0,1\}^m$ such that $w \notin G$.*

We next show haw to generalize the scheme for **NQRP** to a scheme for $\mathcal{A}_{R,G}$.

Distribution. The dealer chooses at random $m-1$ random elements $z_0, \ldots, z_{m-2} \in R$ and an additional random element $r \in G$. Define $z_{m-1} \overset{\text{def}}{=} -\sum_{i=0}^{m-2} z_i$. The shares of the parties are specified in Table A.1.

Table A.1
A secret-sharing scheme for $\mathcal{A}_{R,G}$.

	$s = 0$	$s = 1$
P_0^b, where $b \in \{0,1\}$	$r + z_i$	$br + z_i$
P_i^b, where $1 \leq i < m$, $b \in \{0,1\}$	z_i	$2^i br + z_i$

The reconstruction is similar to the scheme for **NQRP**, where if $B = B_w$ for some $w \notin G$, then $s = 0$ iff $\text{SUM}_w \in G$. The correctness of this rule follows from the fact that if $w \notin G$ and $b \in G$, then $w * b \notin G$.

For the security, we only consider the case where $C = B_w$ for some $w \in G$. (The first case is identical to the scheme for **NQRP**.) In this case we claim that, regardless of the value of the secret, the vector-share of the parties in C is a random vector such that $\text{SUM}_w \in G$. This is clearly true when $s = 0$. When $s = 1$, the sum SUM_w is $r \sum_{i=0}^{m-1} w_i 2^i = rw$, and since r is a random element of G and w has an inverse in G, the product is a random element of G.

We next show a few examples of access structures.

Example A.2. Let N be a positive integer, $R = \langle \mathcal{Z}_N, +, *\rangle$, and $G = \langle \mathcal{Z}_N^*, *\rangle$. In this case, an efficient linear scheme for $\mathcal{A}_{R,G}$ exists (see footnote 17). A quasi-linear scheme for this access structure is described in Example 5.7. However, both the linear and the quasi-linear schemes require knowing the factorization of N. The nonlinear scheme does not require knowledge of the factorization, and all the computations involved are efficient.

Example A.3. Let p be a prime, $R = \langle \mathcal{Z}_{p^2}, +, *\rangle$,

$$B = \left\{w \in \mathcal{Z}_{p^2} : w^{p-1} \equiv 1 \bmod p^2\right\},$$

and $G = \langle B, *\rangle$. In this case we do not know if there is a quasi-linear scheme for $\mathcal{A}_{R,G}$, or even if $\mathcal{A}_{R,G}$ is in NC.

[18] We do not even need all the properties of these algebraic structures.

Appendix B. Explicit estimates implied by the ERH. The next theorem gives explicit bounds on the error term in the approximation of the distribution of the primes.

THEOREM B.1. *Let* $\mathrm{li}(x) \stackrel{def}{=} \int_2^x \frac{dt}{\log t}$. *If the ERH holds, then for* $x \geq 2657$

$$\frac{x}{\log x + 2} < \pi(x) \ < \ \frac{x}{\log x - 4}, \qquad [3, \text{Theorem 8.8.1}] \tag{B.1}$$

and

$$|\pi(x) - \mathrm{li}(x)| \leq \sqrt{x}\log x / 8\pi. \qquad [3, \text{p. } 249] \tag{B.2}$$

Moreover, if the ERH holds, $u \leq x$, *and* $\gcd(w, u) = 1$, *then*

$$\begin{aligned} \left|\pi(x, u, w) - \frac{\mathrm{li}(x)}{\varphi(u)}\right| &\leq \sqrt{x}(\log x + 2\log u) \\ &\leq 3\sqrt{x}\log x. \qquad [3, \text{Theorem 8.8.18}] \end{aligned} \tag{B.3}$$

We next show how we derive Theorem 4.9 from Theorem B.1. That is, we prove that if the ERH holds and $\gcd(w, u) = 1$, then for large x and x', where $u \leq x' \leq \sqrt{x}$,

$$\left|\frac{\pi(x, u, w) - \pi(x', u, w)}{\pi(x) - \pi(x')} - \frac{1}{\varphi(u)}\right| = O\left(\frac{\log^2 x}{\sqrt{x}}\right).$$

First, by (B.1) and since $\pi(x') \leq x' \leq \sqrt{x}$,

$$\pi(x) - \pi(x') > \frac{x}{\log x + 2} - \sqrt{x} \ > \ \frac{x}{2\log x}. \tag{B.4}$$

Second, by (B.3), (B.2), and since $\pi(x') \leq x' \leq \sqrt{x}$,

$$\pi(x, u, w) < \frac{\mathrm{li}(x)}{\varphi(u)} + O(\sqrt{x}\log x) \ < \ \frac{\pi(x) - \pi(x')}{\varphi(u)} + O(\sqrt{x}\log x). \tag{B.5}$$

Therefore, by (B.5) and (B.4),

$$\begin{aligned} \frac{\pi(x, u, w) - \pi(x', u, w)}{\pi(x) - \pi(x')} < \frac{\pi(x, u, w)}{\pi(x) - \pi(x')} \ &\leq \ \frac{1}{\varphi(u)} + O\left(\frac{\sqrt{x}\log x}{\pi(x) - \pi(x')}\right) \\ &\leq \frac{1}{\varphi(u)} + O\left(\frac{\log^2 x}{\sqrt{x}}\right). \end{aligned}$$

On the other hand, by (B.3), since $\pi(x', u, w) < x' \leq \sqrt{x}$, and by (B.2), and (B.1),

$$\begin{aligned} \pi(x, u, w) - \pi(x', u, w) &> \frac{\mathrm{li}(x)}{\varphi(u)} - O(\sqrt{x}\log x) - \sqrt{x} \\ &> \frac{\pi(x) - \sqrt{x}\log x/8\pi - \pi(x')}{\varphi(u)} - O(\sqrt{x}\log x) \\ &> \frac{\pi(x) - \pi(x')}{\varphi(u)} - O(\sqrt{x}\log x). \end{aligned} \tag{B.6}$$

Thus, by (B.6) and (B.4)

$$\frac{\pi(x, u, w) - \pi(x', u, w)}{\pi(x) - \pi(x')} > \frac{1}{\varphi(u)} - O\left(\frac{\sqrt{x}\log x}{\pi(x) - \pi(x')}\right) \ \geq \ \frac{1}{\varphi(u)} - O\left(\frac{\log^2 x}{\sqrt{x}}\right).$$

Acknowledgment. We wish to thank Eric Allender, Daniel Berend, Yuval Ginosar, and Dieter van Melkebeek for helpful discussions and pointers.

REFERENCES

[1] L. M. Adleman and K. Kompella, *Using smoothness to achieve parallelism*, in Proceedings of the 20th Annual ACM Symposium on the Theory of Computing, 1988, pp. 528–538.

[2] L. Babai, A. Gál, and A. Wigderson, *Superpolynomial lower bounds for monotone span programs*, Combinatorica, 19 (1999), pp. 301–319.

[3] E. Bach and J. Shalit, *Algorithmic Number Theory*, Vol. 1. Efficient Algorithms, MIT Press, Cambridge, MA, 1996.

[4] P. Beguin and A. Cresti, *General short computational secret sharing schemes*, in Advances in Cryptology – EUROCRYPT '95, L. C. Guillou and J. J. Quisquater, eds., Lect. Notes in Comput. Sci. 921, Springer-Verlag, Berlin, 1995, pp. 194–208.

[5] R. Beigel and B. Fu, *Circuits over PP and PL*, J. Comput. Syst. Sci., 60 (2000), pp. 422–441. Preliminary version in the proceedings of the 12th Annual IEEE Conference on Computational Complexity, 1997, pp. 24–35.

[6] A. Beimel, *Secure Schemes for Secret Sharing and Key Distribution*, Ph.D. thesis, Technion–Israel Institute of Technology, Haifa, Israel, 1996.

[7] A. Beimel, A. Gál, and M. Paterson, *Lower bounds for monotone span programs*, Comput. Complexity, 6 (1996/1997), pp. 29–45. Conference version: FOCS '95.

[8] A. Beimel and E. Weinreb, *Separating the power of monotone span programs over different fields*, in Proceeding of the 44th Annual IEEE Symposium on Foundations of Computer Science, Cambridge, MA, 2003, pp. 428–437.

[9] M. Ben-Or, S. Goldwasser, and A. Wigderson, *Completeness theorems for noncryptographic fault-tolerant distributed computations*, in Proceedings of the 20th Annual ACM Symposium on the Theory of Computing, Chicago, IL, 1988, pp. 1–10.

[10] J. Benaloh and J. Leichter, *Generalized secret sharing and monotone functions*, in Advances in Cryptology – CRYPTO '88, S. Goldwasser, ed., Lect. Notes in Comput. Sci. 403, Springer-Verlag, Berlin, 1990, pp. 27–35.

[11] J. Benaloh and S. Rudich, *Private communication*, 1989.

[12] S. J. Berkowitz, *On computing the determinant in small parallel time using a small number of processors*, Inform. Process. Lett., 18 (1984), pp. 147–150.

[13] M. Bertilsson and I. Ingemarsson, *A construction of practical secret sharing schemes using linear block codes*, in Advances in Cryptology – AUSCRYPT '92, J. Seberry and Y. Zheng, eds., Lect. Notes Comput. Sci. 718, Springer-Verlag, Berlin, 1993, pp. 67–79.

[14] G. R. Blakley, *Safeguarding cryptographic keys*, in Proceedings of the 1979 AFIPS National Computer Conference, R. E. Merwin, J. T. Zanca, and M. Smith, eds., Vol. 48 of AFIPS Conference Proceedings, AFIPS Press, Arlington, VA, 1979, pp. 313–317.

[15] C. Blundo, A. De Santis, L. Gargano, and U. Vaccaro, *On the information rate of secret sharing schemes*, Theoret. Comput. Sci., 154 (1996), pp. 283–306.

[16] D. Boneh and M. Naor, *Timed commitments*, in Advances in Cryptology – CRYPTO 2000, M. Bellare, ed., Lect. Notes Comput. Sci. 1880, Springer-Verlag, Berlin, 2000, pp. 236–254.

[17] A. Borodin, J. von zur. Gathen, and J. Hopcroft, *Fast parallel matrix and GCD computations*, Inform. Control, 52 (1982), pp. 241–256.

[18] E. F. Brickell, *Some ideal secret sharing schemes*, J. Combin. Math. Combin. Comput., 6 (1989), pp. 105–113.

[19] E. F. Brickell and D. M. Davenport, *On the classification of ideal secret sharing schemes*, J. Cryptology, 4 (1991), pp. 123–134.

[20] E. F. Brickell and D. R. Stinson, *Some improved bounds on the information rate of perfect secret sharing schemes*, J. Cryptology, 5 (1992), pp. 153–166.

[21] G. Buntrock, C. Damm, U. Hertrampf, and C. Meinel, *Structure and importance of the logspace-mod class*, Math. Systems Theory, 25 (1992), pp. 223–237.

[22] R. M. Capocelli, A. De Santis, L. Gargano, and U. Vaccaro, *On the size of shares for secret sharing schemes*, J. Cryptology, 6 (1993), pp. 157–168.

[23] D. Chaum, C. Crépeau, and I. Damgård, *Multiparty unconditionally secure protocols*, in Proceedings of the 20th Annual ACM Symposium on the Theory of Computing, Chicago, IL, 1988, pp. 11–19.

[24] B. Chor and O. Goldreich, *An improved parallel algorithm for integer GCD*, Algorithmica, 5 (1990), pp. 1–10.

[25] S. A. Cook, *A taxonomy of problems with fast parallel algorithms*, Inform. Control, 64 (1985), pp. 2–22.

[26] R. Cramer, I. Damgård, and S. Dziembowski, *On the complexity of verifiable secret sharing and multiparty computation*, in Proceedings of the 32nd Annual ACM Symposium on the Theory of Computing, New York, 2000, pp. 325–334.

[27] R. Cramer, I. Damgård, and U. Maurer, *General secure multi-party computation from any linear secret-sharing scheme*, in Advances in Cryptology – EUROCRYPT 2000, B. Preneel, ed., Lect. Notes Comput. Sci. 1807, Springer-Verlag, Berlin, 2000, pp. 316–334.

[28] L. Csirmaz, *The dealer's random bits in perfect secret sharing schemes*, Studia Sci. Math. Hungar., 32 (1996), pp. 429–437.

[29] L. Csirmaz, *The size of a share must be large*, J. Cryptology, 10 (1997), pp. 223–231.

[30] Y. Desmedt and Y. Frankel, *Shared generation of authenticators and signatures*, in Advances in Cryptology – CRYPTO '91, J. Feigenbaum, ed., Lect. Notes Comput. Sci. 576, Springer-Verlag, Berlin, 1992, pp. 457–469.

[31] Y. Desmedt and Y. Frankel, *Homomorphic zero-knowledge threshold schemes over any finite abelian group*, SIAM J. Discrete Math., 7 (1994), pp. 667–679.

[32] M. v. Dijk, *On the information rate of perfect secret sharing schemes*, Des. Codes Cryptogr., 6 (1995), pp. 143–169.

[33] M. v. Dijk, *A linear construction of secret sharing schemes*, Des. Codes Cryptogr., 12 (1997), pp. 161–201.

[34] S. M. Eikenberry and J. P. Sorenson, *Efficient algorithms for computing the Jacobi symbol*, J. Symb. Comput., 26 (1998), pp. 509–523.

[35] S. Fehr, *Span Programs Over Rings and How to Share a Secret from a Module*, Master's thesis, ETH Zurich, Zurich, Switzerland, 1998.

[36] U. Feige, J. Kilian, and M. Naor, *A minimal model for secure computation*, in Proceedings of the 26th Annual ACM Symposium on the Theory of Computing, Montreal, 1994, pp. 554–563.

[37] A. Gál, *A characterization of span program size and improved lower bounds for monotone span programs*, in Proceedings of the 30th Annual ACM Symposium on the Theory of Computing, Dallas, 1998, pp. 429–437.

[38] L. Goldschlager, *The monotone and planar circuit value problem is complete for P*, SIGACT News, 9 (1977), pp. 25–27.

[39] S. Goldwasser and S. Micali, *Probabilistic encryption*, J. Comput. System Sci., 28 (1984), pp. 270–299.

[40] D. M. Gordon, *A survey of fast exponentiation methods*, J. Algorithms, 27 (1998), pp. 129–146.

[41] J. Håstad, *The shrinkage exponent of De Morgan formulas is* 2, SIAM J. Comput., 27 (1998), pp. 48–64.

[42] Y. Ishai and E. Kushilevitz, *Private simultaneous messages protocols with applications*, in 5th Israel Symposium on Theory of Computing and Systems, Ramat-Gan, Israel, 1997, pp. 174–183.

[43] M. Ito, A. Saito, and T. Nishizeki, *Secret sharing schemes realizing general access structure*, in Proceedings of the IEEE Global Telecommunication Conference, Globecom 87, 1987, pp. 99–102. Journal version: *Multiple Assignment Scheme for Sharing Secret., J. Cryptology*, 6 (1993), pp. 15–20.

[44] R. Kannan, G. L. Miller, and L. Rudolph, *Sublinear parallel algorithm for computing the greatest common divisor of two integers*, SIAM J. Comput., 16 (1987), pp. 7–16.

[45] M. Karchmer and A. Wigderson, *On span programs*, in Proceedings of the 8th Annual IEEE Structure in Complexity Theory, San Diego, 1993, pp. 102–111.

[46] H. Krawczyk, *Secret sharing made short*, in Advances in Cryptology – CRYPTO '93, D. R. Stinson, ed., Lect. Notes Comput. Sci. 773, Springer-Verlag, Berlin, 1994, pp. 136–146.

[47] K. M. Martin, *New secret sharing schemes from old*, J. Combin. Math. Combin. Comput., 14 (1993), pp. 65–77.

[48] E. Martinez-Moro, J. Mozo-Fernandez, and C. Munuera, *Compounding secret sharing schemes*, Australas. J. Combin., 30 (2004), pp. 277–290.

[49] K. Mulmuley, *A fast parallel algorithm to compute the rank of a matrix over an arbitrary field*, Combinatorica, 7 (1987), pp. 101–104.

[50] M. O. Rabin, *Randomized Byzantine generals*, in Proceedings of the 24th IEEE Symposium on Foundations of Computer Science, Boston, 1983, pp. 403–409.

[51] A. Renvall and C. Ding, *A nonlinear secret sharing scheme*, in ACISP: Information Security and Privacy: Australasian Conference, Lect. Notes Comput. Sci. 1172, Springer-Verlag, Berlin, 1996, pp. 56–66.

[52] A. Shamir, *How to share a secret*, Comm. ACM, 22 (1979), pp. 612–613.

[53] G. J. Simmons, *An introduction to shared secret and/or shared control and their application*, in Contemporary Cryptology, The Science of Information Integrity, G. J. Simmons, ed., IEEE Press, Piscataway, NJ, 1992, pp. 441–497.

[54] G. J. Simmons, W. Jackson, and K. M. Martin, *The geometry of shared secret schemes*,

Bull. Inst. Combin. Appl., 1 (1991), pp. 71–88.

[55] J. Simonis and A. Ashikhmin, *Almost affine codes*, Des. Codes Cryptogr., 14 (1998), pp. 179–197.

[56] J. Sorenson, *Two fast GCD algorithms*, J. Algorithms, 16 (1994), pp. 110–144.

[57] P. M. Spira, *On time hardware tradeoffs for Boolean functions*, in Proceedings of the 4th Annual Hawaii International Symposium on System Sciences, Western Periodicals Co., North Hollywood, 1971, pp. 525–527.

[58] D. R. Stinson, *An explication of secret sharing schemes*, Des. Codes Cryptogr., 2 (1992), pp. 357–390.

[59] D. R. Stinson, *New general lower bounds on the information rate of secret sharing schemes*, in Advances in Cryptology – CRYPTO '92, E. F. Brickell, ed., Lect. Notes Comput. Sci. 740, Springer-Verlag, Berlin, 1993, pp. 168–182.

[60] D. R. Stinson and J. L. Massey, *An infinite class of counterexamples to a conjecture concerning nonlinear resilient functions*, J. Cryptology, 8 (1995), pp. 167–173.

[61] D. R. Stinson and S. A. Vanstone, *A combinatorial approach to threshold schemes*, SIAM J. Discrete Math., 1 (1988), pp. 230–236.

[62] A. C. Yao, Unpublished manuscript, Presented at Oberwolfach (Oberwolfach, Germany) and DIMACS (Piscataway, NJ) workshops, 1989.

SIAM J. Discrete Math.
Vol. 19, No. 2, pp. 281–293

CIRCULAR DISTANCE TWO LABELING AND THE λ-NUMBER FOR OUTERPLANAR GRAPHS*

DAPHNE DER-FEN LIU[†] AND XUDING ZHU[‡]

Abstract. Let G be a graph. A circular distance two labeling with span k is a function $f : V(G) \to \{0, 1, 2, \ldots, k-1\}$ such that (1) $2 \le |f(u) - f(v)| \le k - 2$ if u and v are adjacent and (2) $f(u) \neq f(v)$ if u and v are of distance two apart. We denote by $\lambda_c(G)$ the smallest span of a circular distance two labeling for G. Let $\Delta(G)$ be the maximum degree of G. We prove, for any outerplanar graph G with $\Delta(G) \ge 15$, $\lambda_c(G) = \Delta(G) + 3$. It is also shown that there exist outerplanar graphs G with $\Delta(G) = 2, 3, 4, 5$ for which $\lambda_c(G) = \Delta(G) + 4$. Moreover, we prove that $\lambda_c(G) \le \Delta(G) + 5$ for any triangulated outerplanar graph, $\lambda_c(G) \le \Delta(G) + 7$ for any outerplanar graph, and $\lambda_c(G) \le \Delta(G) + 4$ for any outerplanar graph with $\Delta(G) \ge 11$. Immediate consequences of our results include that $\lambda(G) \le \Delta(G) + 2$ for any outerplanar graphs with $\Delta(G) \ge 15$, where $\lambda(G)$ is the minimum k of a k-$L(2,1)$-labeling (or distance two labeling) for G.

Key words. circular distance two labeling, distance two labeling, $L(2,1)$-labeling, outerplanar graphs

AMS subject classification. 05C15

DOI. 10.1137/S0895480102414296

1. Introduction.

Distance two labeling (or $L(2,1)$-labeling) is motivated by the channel assignment problem (cf. [4]). The task is to assign one nonnegative integral channel to each of the given transmitters or stations so that interference is avoided and the span of all the channels used is minimized.

Suppose that we are dealing with two levels of interference—major and minor. Major interference occurs between two close transmitters. To avoid it, the difference of the channels assigned to such a pair of transmitters must be at least 2. Minor interference occurs between two transmitters that share a common close neighbor. To avoid it, the difference of the channels assigned to such a pair of transmitters must be at least 1.

Let $G = (V, E)$ be the graph where each vertex represents a transmitter, and two vertices are adjacent if the corresponding transmitters are close. The above channel assignment corresponds to an $L(2,1)$-*labeling* of G, which is defined to be a function $f : V(G) \to \{0, 1, 2, \ldots\}$ such that the following are satisfied, where $d_G(u, v)$ denotes the distance between u and v in G:

- $|f(x) - f(y)| \ge 2$ if $d_G(x, y) = 1$; and
- $|f(x) - f(y)| \ge 1$ if $d_G(x, y) = 2$.

The *span* of f is defined as $\text{span}(f) = \max_{x \in V} f(x) - \min_{x \in V} f(x)$. If $\text{span}(f) = k$, then f is called a k-$L(2,1)$-labeling. Without loss of generality, for convenience we assume that $\min_{x \in V} f(x) = 0$, and hence $\max_{x \in V} f(x) = \text{span}(f)$. The numbers

*Received by the editors September 11, 2002; accepted for publication (in revised form) July 21, 2004; published electronically October 20, 2005.

http://www.siam.org/journals/sidma/19-2/41429.html

[†]Department of Mathematics, California State University, Los Angeles, Los Angeles, CA 90032 (dliu@calstatela.edu). This author's research was supported in part by National Science Foundation grants DMS 9805945 and DMS 0302456.

[‡]Department of Applied Mathematics, National Sun Yat-sen University, Kaohsiung, Taiwan 80424 (zhu@ibm18.math.nsysu.edu.tw). This author's research was supported in part by National Science Council grant NSC 91-2115-M-110-003.

$0, 1, 2, \ldots, k$ are called *colors* (or labels). The λ-*number* of G, denoted by $\lambda(G)$, is the minimum k such that G admits a k-$L(2,1)$-labeling.

A *circular distance two labeling* with span k (or a k-$L_c(2,1)$-labeling) of a graph G is a function, $f : V(G) \to \{0, 1, 2, \ldots, k-1\}$ such that the following are satisfied:

- $|f(x) - f(y)|_k \geq 2$ if $d_G(x, y) = 1$, and
- $|f(x) - f(y)|_k \geq 1$ if $d_G(x, y) = 2$,

where $|x-y|_k$, the *modular k circular difference* between x, y, is defined as $|x-y|_k = \min\{|x-y|, k-|x-y|\}$. The *circular λ-number* of G, denoted by $\lambda_c(G)$, is the smallest k such that G admits a k-$L_c(2,1)$-labeling. Circular distance two labelings and the values of $\lambda_c(G)$ for different families of graphs have been studied in [5, 7, 8, 9, 10].

By definition, every $(k+1)$-$L_c(2,1)$-labeling is a k-$L(2,1)$-labeling, and every k-$L(2,1)$-labeling is a $(k+2)$-$L_c(2,1)$-labeling. Therefore, we have

$$\lambda(G) + 1 \leq \lambda_c(G) \leq \lambda(G) + 2. \tag{1.1}$$

The colors in a circular distance two labeling are *symmetric* in the following sense. Let f be a k-$L_c(2,1)$-labeling of G. Then, for any $i \in \{0, 1, 2, \ldots, k-1\}$, the function defined by $f^*(u) = f(u) - i \pmod{k}$ is also a k-$L_c(2,1)$-labeling for G. The colors in a distance two labeling do not have this property. For instance, the star $K_{1,n}$ has $\lambda(G) = n+1$, and any optimal $L(2,1)$-labeling must assign to the center vertex either 0 or $n+1$. This kind of asymmetry in colors sometimes causes difficulties in discussion. In this article, we take advantage of the symmetry of colors in a circular distance two labeling to explore the value of $\lambda_c(G)$, which, by (1.1), gives good bounds for the λ-number of G.

The circular λ-number of graphs is closely related to the circular chromatic number of edge weighted graphs, a notion introduced by Mohar [11]. An *edge weighted graph* with vertex set V is a pair $G = (V, A)$, where $A : V \times V \to R^+ \cup \{0\}$ is a weight assignment to the ordered pairs $(u, v) \in V \times V$. For every (u, v), we write $a_{uv} = A(u, v)$. For a positive real number p, denote by $S_p \subset R^2$ the circle with perimeter p centered at the origin of R^2. For any $x, y \in S_p$, let $l(x, y)$ denote the length of the arc from x to y on S_p, in the clockwise direction. A *circular p-coloring* of $G = (V, A)$ is a function $c : V \to S_p$ such that $l(c(u), c(v)) \geq a_{uv}$ for every $(u, v) \in V \times V$. The *circular chromatic number* $\chi_c(G)$ of the graph $G = (V, A)$ is the infimum of all real numbers p for which there exists a circular p-coloring of G. If $a_{uv} = a_{vu}$ for every $u, v \in V$, then the weights are called *symmetric*.

For any undirected graph $G(V, E)$, we construct a symmetric edge weighted graph $G(2,1) = (V, A)$ by the following: (1) for each $uv \in E(G)$, let $a_{uv} = a_{vu} = 2$; (2) if u' and v' are distance two apart in G, then $a_{u'v'} = a_{v'u'} = 1$; and (3) $a_{uv} = 0$ for all other pairs (u, v). It is not hard to verify that the following holds for any graph G [9]:

$$\lambda_c(G) = \lceil \chi_c(G(2,1)) \rceil.$$

Determining $\lambda(G)$ is an NP-complete problem, even restricted to special classes of graphs, such as graphs with diameter 2 [3], planar graphs, bipartite graphs, chordal graphs, or split graphs (cf. [1]). Research on the parameter $\lambda(G)$ has been concentrated on finding good upper bounds for $\lambda(G)$. Denote by $\Delta(G)$ the maximum degree of G, or Δ when G is clear in the context. It is easy to see that for any graph G, $\lambda(G) \geq \Delta + 1$ and $\lambda_c(G) \geq \Delta + 3$. For general upper bounds of $\lambda(G)$, it was shown in [3] that for any G, $\lambda(G) \leq \Delta^2 + 2\Delta$. Chang and Kuo [2] improved this bound by showing that $\lambda(G) \leq \Delta^2 + \Delta$ (where the proof actually shows that $\lambda_c(G) \leq \Delta^2 + \Delta + 1$). Recently, using the notion of list coloring, Král' and Škrekovski [6] further improved this bound to $\lambda(G) \leq \Delta^2 + \Delta - 1$. A still open conjecture [3] states that $\lambda(G) \leq \Delta^2$ for any graph G. For special classes of graphs, better upper bounds are known. Below we list some of the known results on $\lambda(G)$ for some families of graphs.

Graphs	$\lambda(G)$	Reference
Trees	$\Delta + 1$ or $\Delta + 2$	Chang and Kuo [2]
Chordal	$\leq \frac{1}{4}(\Delta + 3)^2$	Sakai [13]
Diameter two	$\leq \Delta^2$	Griggs and Yeh [3]
Planar	$\leq 2.5\Delta + 90$	Molloy and Salavatipour [12]
Outerplanar (OP)	$\leq \Delta + 8$	Bodlaender et al. [1]
Triangulated OP	$\leq \Delta + 6$	Bodlaender et al. [1]

A graph is *outerplanar* if it can be embedded in the plane in such a way that all the vertices lie on the infinite face. We call such an embedding an *outerplane graph*. An outerplanar graph is *triangulated* if it is 2-connected (i.e., without cut-vertices) and can be drawn as an outerplane graph such that each finite face is a triangle. In searching for the λ-number of outerplanar graphs, Bodlaender et al. [1] proposed the following.

Conjecture 1. *For any outerplanar graph G, $\lambda(G) \leq \Delta + 2$.*

The main result of this article is the following.

Theorem 1. *For any outerplanar graph G with $\Delta \geq 15$, $\lambda_c(G) = \Delta + 3$.*

By (1.1), an immediate consequence of Theorem 1 is the confirmation of Conjecture 1 for outerplanar graphs with large maximum degree.

Corollary 2. *For any outerplanar graph G with $\Delta \geq 15$, $\lambda(G) \leq \Delta + 2$.*

For outerplanar graphs with smaller maximum degree, we prove the following two results.

Theorem 3. *Suppose G is an outerplanar graph. Then $\lambda_c(G) \leq \Delta(G) + 7$. Moreover, if G is triangulated, then $\lambda_c(G) \leq \Delta(G) + 5$.*

Theorem 4. *If G is an outerplanar graph and $\Delta(G) \geq 11$, then $\lambda_c(G) \leq \Delta(G) + 4$.*

Combining Theorem 3 with (1.1), we are able to improve the bounds of the λ-number for outerplanar graphs obtained in [1].

Note that the condition $\Delta(G) \geq 15$ in Theorem 1 cannot be simply removed. In the last section of this article, we demonstrate the existence of outerplanar graphs G with $\Delta(G) = 2, 3, 4, 5$ and $\lambda_c(G) = \Delta(G) + 4$.

2. Structure of outerplanar graphs. Let G be an outerplanar graph. Then G can be transformed into a triangulated outerplane graph G_T by adding some edges. We call G_T a *triangulation* of G. There may exist many triangulations of G; however, we denote by G_T an arbitrary but fixed triangulation.

Let G be a triangulated outerplane graph. We define a *level function* l on $V(G)$, by recursion, such that $l(u) \neq l(v)$ if $u \sim v$. Initially, choose an edge $e = u_1u_2$ on the infinite face and let $l(u_1) = 1$ and $l(u_2) = 2$; we call e the *root edge* and u_1, u_2 the *root vertices*. Let $X = \{v \in V(G) : l(v) \text{ is defined}\}$. While $X \neq V(G)$, choose a triangle (u, v, w) such that $v, w \in X$ and $u \in V(G) - X$. Assume $l(v) > l(w)$ (since $v \sim w$, by inductive hypothesis $l(v) \neq l(w)$). Let $l(u) = l(v) + 1$. The vertices w, v are called the *major parent* and the *minor parent* of u and are denoted by $w = f(u)$ and $v = m(u)$, respectively. It is easy to verify that, at each step, the subgraph $G[X]$ of G induced by X is a triangulated outerplane graph. This implies that if, at some step, $u \in V(G) - X$ is contained in a triangle (u, v, w) such that $v, w \in X$, then the triangle is unique. Hence, for any nonroot vertex u, the functions $l(u)$, $m(u)$ and $f(u)$ are well defined. The following lemma follows from the definitions.

LEMMA 5. *If u and $m(u)$ are nonroot vertices, then $f(u) \in \{m(m(u)), f(m(u))\}$. If $u' \neq u$ are two nonroot vertices, then $\{f(u), m(u)\} \neq \{f(u'), m(u')\}$.*

If v is a parent of u, then u is called a *child* of v. If v is the major (respectively, minor) parent of u, then u is called a major (respectively, minor) child of v. If $m(u) = m(u')$, then u and u' are *siblings.* Note that a vertex may have many children. However, the following lemma shows that each vertex has at most one sibling and at most two children of the same level.

LEMMA 6. *Suppose G is a triangulated outerplanar graph with level function l. Let v be a vertex and i a positive integer. Then v has at most two children u with $l(u) - l(v) = i$. In particular, v has at most two minor children and at most one sibling.*

Proof. Let $W_i = \{x : x \text{ is a child of } v \text{ with } l(x) = l(v) + i\}$. We prove by induction on i that $|W_i| \leq 2$. If $u \in W_1$, then $m(u) = v$. If v is a root vertex, then it follows from the definition that $|W_1| \leq 1$. If v is a nonroot vertex, by Lemma 5, $f(u) \in \{f(v), m(v)\}$. Hence, the parents of u are either $\{v, f(v)\}$ or $\{v, m(v)\}$. By Lemma 5, there is at most one vertex whose parents are $\{v, f(v)\}$ and at most one vertex whose parents are $\{v, m(v)\}$. Therefore $|W_1| \leq 2$. Assume $|W_k| \leq 2$ for some $k \geq 1$. If $u \in W_{k+1}$, then $v = f(u)$ and $m(u) \in W_k$. Since $|W_k| \leq 2$, it follows from Lemma 5 that $|W_{k+1}| \leq 2$.

If u' and u are siblings, then u and u' are both minor children of $m(u) = m(u')$, of level $l(m(u)) + 1$. As $m(u)$ has at most two children with level $l(m(u)) + 1$, we conclude that each vertex has at most one sibling. □

If G is a nontriangulated outerplanar graph, then we define the level function l on a triangulation G_T of G, and view l as a function on G. Similarly, parents, children and siblings are defined according to l in the same manner. Note that as G is nontriangulated, a vertex u may not be adjacent to its parents.

Next, we define a lexicographic ordering $\prec$ on $V(G)$ by the following:

- $u_1 \prec u_2$.
- If $m(u) \prec m(u')$, then $u \prec u'$. If $m(u) = m(u')$, then $u \prec u'$ if and only if $f(u) \prec f(u')$.

By Lemma 5, $\prec$ is a linear ordering on $V(G)$. Throughout this article, we write $V(G)$ as $V(G) = \{v_1, v_2, \ldots, v_n\}$, where $v_i \prec v_j$ if and only if $i < j$. In particular, $v_1 = u_1, v_2 = u_2$ are the two root vertices.

Let t be an integer, $1 \leq t \leq n$. Denote $V_t = \{v_1, v_2, \ldots, v_t\}$. Let $w \in V$. We denote the number of neighbors of w in V_t by $s[w, t]$, that is,

$$s[w, t] = |\{v_j : j \leq t, v_j \sim w\}|.$$

Observe that for any nonroot vertex $w = v_b$ of an outerplanar graph G, if $f(w) = v_i$ and $m(w) = v_j$, then $i < j < b$, $s[w, i] \leq 1$, and $s[w, j] = s[w, b] \leq 2$. Moreover, we have the following.

LEMMA 7. *Let G be an outerplanar graph and G_T a triangulation of G. Let $w \in V(G)$. Suppose $w \sim v_t$ in G_T.*

(1) *If $s[w, t] \geq 5$, then $w = f(v_t)$.*
(2) *If $s[w, t] \geq 7$, then $f(m(v_t)) = w$.*
(3) *If $s[w, t] \geq 9$, then $f(m(m(v_t))) = w$.*
(4) *If $m(v_t) = v_{t'}$, then $s[w, t] - s[w, t'] \leq 2$.*
(5) *If $w = f(v_l) = f(v_t)$, and $v_l \sim v_t$ for some l, then $|s[w, t] - s[w, l]| \leq 2$.*

Proof. The neighbors of w in G_T, in the ordering $\prec$, are, first, the parents of w; second, the minor children of w; and finally, the major children of w. Note that w has only two parents and, by Lemma 6, at most two minor children. Hence, if $s[w,t] \geq 5$, then v_t must be a major child of w; i.e., $w = f(v_t)$. So (1) holds.

The rest of the lemma can be proved similarly, and we omit the details. □

3. Proofs of Theorems 3 and 4. Suppose G is an outerplanar graph with vertex set $V = \{v_1, v_2, \ldots, v_n\}$, ordered as in section 2. To prove Theorems 1, 3, and 4, it suffices to find a k-$L_c(2,1)$-labeling for G, with the corresponding desired value of k. We regard the colors $\{0,1,2,\ldots,k-1\}$ on a circular palette, and all calculations are taken modulo k. Let C be a proper subset of colors on this color palette. A *segment* of C is a maximal interval of consecutive colors of C, i.e., a set I of colors of the form $I = \{j, j+1, \ldots, l\}$ such that $I \subset C$ and $j-1, l+1 \notin C$. The colors between two consecutive segments are called a *gap* of C. As we are working on a circular color palette (i.e., modulo k), the number of gaps is the same as the number of segments.

Let C be a proper subset of $\{0,1,2,\ldots,k-1\}$. A color j is called *attaching* to C if $j+1$ or $j-1$ belongs to C. A color j is called a *filling* color of C if both $j+1$ and $j-1$ belong to C. Denote by $A(C)$ and $F(C)$, respectively, the set of attaching colors and the set of filling colors of C.

PROPOSITION 8. *Let C be a proper subset of* $\{0,1,2,\ldots,k-1\}$ (mod k).

(1) *If $x \in F(C) - C$, then $\{x\}$ is a (singleton) gap of C.*

(2) *If $x \in C - A(C)$, then $\{x\}$ is a (singleton) segment of C.*

For all the proofs of Theorems 1, 3, and 4, we define a sequential labeling for G, according to the ordering $v_1, v_2, \ldots, v_n$.

Suppose that ϕ is a *partial labeling* for V_{t-1} (i.e., ϕ is an assignment of colors to V_{t-1} which can be extended to a k-$L_c(2,1)$-labeling for G). For any $b \geq t$, a color j is *legal* for v_b if for all $u \in V_{t-1}$, the following hold:

- if $u \sim v_b$, then $j \notin \{\phi(u), \phi(u) \pm 1\}$; and
- if $d_G(u, v_b) = 2$, then $j \neq \phi(u)$.

At each step, we extend ϕ from V_{t-1} to V_t by assigning a legal color to v_t. A color is *forbidden* for v_t if it is not legal for v_t. We denote by $\mathrm{Forb}(v_t)$ the set of forbidden colors for v_t.

For $u \in V_{t-1}$, set

$$C[u, t-1] = \{\phi(u), \phi(u)+1, \phi(u)-1\} \cup \{\phi(v_j) : j \leq t-1, v_j \sim u\}.$$

LEMMA 9. *Let G be an outerplanar graph. Suppose ϕ is a partial labeling for V_{t-1}. Then the following hold:*

(1) $\mathrm{Forb}(v_t) \subseteq C[m(v_t), t-1] \cup C[f(v_t), t-1]$.

(2) $C[m(v_t), t-1] \subseteq \{\phi(m(v_t)), \phi(m(v_t)) \pm 1, \phi(m(m(v_t))), \phi(f(m(v_t))), \phi(x)\}$, *where x is a possible colored sibling of v_t.*

(3) *If $f(m(v_t)) = f(v_t)$, then the vertex x in* (2) *does not exist, and hence* $|C[m(v_t), t-1]| \leq 5$.

(4) $|\mathrm{Forb}(v_t) - C[f(v_t), t-1]| \leq 5$.

(5) *If $f(m(v_t)) = f(v_t)$, then* $|\mathrm{Forb}(v_t) - C[f(v_t), t-1]| \leq 4$.

(6) *If G is triangulated, then*

$$C[m(v_t), t-1] - C[f(v_t), t-1] \subseteq \{\phi(m(v_t)) \pm 1, \phi(x)\},$$

and $|\mathrm{Forb}(v_t) - C[f(v_t), t-1]| \leq 3$, *where x is a possible colored sibling of v_t.*

(7) *If G is triangulated and $f(m(v_t)) = f(v_t)$, then* $|\mathrm{Forb}(v_t) - C[f(v_t), t-1]| \leq 2$.

Proof. By the ordering $\prec$, the only possible colored neighbors of v_t are $f(v_t)$ and $m(v_t)$. Assume u is a colored vertex for which $d_G(u, v_t) = 2$. Let z be a common neighbor of u and v_t. If z is colored, then $z \in \{f(v_t), m(v_t)\}$ and hence $u \in C[f(v_t), t-1] \cup C[m(v_t), t-1]$. If z is not colored yet, then both u and v_t are parents of z; hence $u \in \{f(v_t), m(v_t)\}$. Therefore (1) is true.

If $v_i \sim m(v_t)$ and $i < t$, then v_i is either a parent of $m(v_t)$ or a sibling of v_t. Hence (2) is true.

If $f(m(v_t)) = f(v_t)$ and x is a sibling of v_t, then $f(x) = m(m(v_t))$ and hence $v_t \prec x$; i.e., x is not colored yet. So (3) holds.

Note that since $f(v_t)$ is also a parent of $m(v_t)$, we have

$$\phi(f(v_t)) \in \{\phi(m(m(v_t))), \phi(f(m(v_t)))\}.$$

By (2), $|C[m(v_t), t-1] - C[f(v_t), t-1]| \leq 5$. This verifies (4).

If $f(m(v_t)) = f(v_t)$, then (5) follows from (1)–(4).

If G is triangulated, then $\{\phi(m(v_t)), \phi(m(m(v_t))), \phi(f(m(v_t)))\} \subseteq C[f(v_t), t-1]$. Hence, (6) follows from (1) and (2). Moreover, if $f(m(v_t)) = f(v_t)$, then (7) follows from (1), (3), and (6). □

Proof of Theorem 3. We first consider the case that G is triangulated. By Lemma 9 (6), $|C[m(v_t), t-1] - C[f(v_t), t-1]| \leq 3$. As $|C[f(v_t), t-1]| \leq \Delta + 2$, by Lemma 9 (1), $|\text{Forb}(v_t)| \leq \Delta + 5$. If we had $\Delta + 6$ colors, then we would always have had a legal color for v_t. However, we are given only $k = \Delta + 5$ colors. So our aim is to reduce the number of forbidden colors of v_t by 1. To accomplish this, we define a sequential coloring scheme such that the following property R1 is satisfied at each step.

R1. If $t \geq 3$, then $\phi(v_t)$ is an attaching color of $C[f(v_t), t-1]$ or $C[m(v_t), t-1]$.

Note that the coloring scheme is based upon the ordering $\prec$; however, if x and y constitute a pair of siblings, then we consider the coloring of x and y *simultaneously*. Observe that it follows from the definition of the ordering $\prec$ that x and y are two *consecutive* vertices in $\prec$.

Initially: Let $\phi(v_1) = 0$, $\phi(v_2) = 2$, and $\phi(v_3) = 4$, so R1 is true.

Inductively: Suppose that ϕ has colored V_{t-1} such that R1 is satisfied at each step, and we want to color the vertex v_t. Assume v_t does not have a sibling. By Lemma 9 (6), we have $|\text{Forb}(v_t)| \leq |C[f(v_t), t-1]| + 2 \leq \Delta + 4$. As we are given $\Delta + 5$ colors, there exists some $j \in A(\text{Forb}(v_t)) - \text{Forb}(v_t)$. Let $\phi(v_t) = j$. Then R1 is satisfied.

Assume v_t has a sibling x. Then we color v_t and x in one step. Let $m(x) = m(v_t) = v_j$ for some $v_j \in V_{t-1}$. Assume that $f(x) = f(v_j)$ and $f(v_t) = m(v_j)$ (the other case, $f(x) = m(v_j)$ and $f(v_t) = f(v_j)$, can be proved similarly). By inductive hypothesis, $\phi(v_j)$ is attaching to $C[m(v_j), j-1]$ or $C[f(v_j), j-1]$.

If $\phi(v_j)$ is attaching to $C[m(v_j), j-1]$, then we first color x by a legal color attaching to $\text{Forb}(x)$. This can be done because x has no colored sibling and hence $|\text{Forb}(x)| \leq \Delta + 4$ (by Lemma 9 (6) and $|C[f(v_t), t-1]| \leq \Delta + 2$). Next, we find a legal color for v_t. Because $m(v_j) = f(v_t)$ and $\phi(v_j)$ is attaching to $C[m(v_j), j-1]$, we conclude that at least one of $\phi(v_j) + 1$ and $\phi(v_j) - 1$ is in $C[f(v_t), t-1]$. By Lemma 9 (6), we have $|C[m(v_t), t-1] - C[f(v_t), t-1]| \leq 2$, and so $|\text{Forb}(v_t)| \leq \Delta + 4$. Hence, there is a legal color for v_t that is attaching to $\text{Forb}(v_t)$, and R1 is satisfied.

If $\phi(v_j)$ is attaching to $C[f(v_j), j-1]$, then we color v_t before x. The discussion is the same as in the previous paragraph. This completes the proof for the existence of a coloring with at most $\Delta + 5$ colors for a triangulated outerplanar graph.

The part of Theorem 3 concerning general outerplanar graphs can be proved similarly, using (4), instead of (6), of Lemma 9. We omit the details. □

Proof of Theorem 4. Let G be an outerplanar graph with $\Delta \geq 11$. Let $k = \Delta + 4$. Similar to the proof of Theorem 3, we give a sequential coloring scheme on the ordering $V(G) = \{v_1, v_2, \ldots, v_n\}$ using colors from the set $\{0, 1, 2, \ldots, k-1\}$. With fewer colors, we need to be more restrictive in bounding the size of $\mathrm{Forb}(v_t)$.

Suppose ϕ is a partial k-$L_c(2,1)$-labeling for V_t, where $t \geq 3$. Let $w = f(v_t)$, and let β be the number of segments in $C[w, t]$. Observe that

$$\beta \leq |C[w,t]| - 2 = s[w,t] + 1. \tag{3.1}$$

We call ϕ a *valid partial labeling* for V_t if R1–R3 in the following hold:

R1. $\beta \leq 5$.

R2. If $s[w,t] \geq 5$, then $\phi(v_t) \in C[w,t]$.

R3. If $s[w,t] \geq 9$, then $\phi(v_t) \in A(C[w,t]) \cap C[w,t]$.

We shall prove that for any $3 \leq t \leq n$, there is a valid partial labeling for V_t.

Initially: Let $\phi(v_1) = 0$, $\phi(v_2) = 2$, and $\phi(v_3) = 4$. Then R1 is true, while R2 and R3 are vacuous.

Inductively: Assume ϕ is a valid partial labeling for V_{t-1}, $t \geq 4$. We extend ϕ to V_t, by assigning a color to v_t, so that ϕ is a valid partial labeling for V_t.

Assume $s[w,t] \leq 4$. Then $|C[w,t-1]| \leq 7$ and $|\mathrm{Forb}(v_t)| \leq 12$ (by Lemma 9 (4)). Let $\phi(v_t) = j$ for some $j \notin \mathrm{Forb}(v_t)$ (j exists because $k = \Delta + 4 \geq 15$). By (3.1), R1 holds. R2 and R3 are vacuous.

Assume $s[w,t] \geq 5$. We consider two cases.

Case 1. $v_t \not\sim w$. Then $s[w,t] = s[w,t-1]$ and $C[w,t-1] = C[w,t]$, regardless of what legal color will be assigned to v_t. By inductive hypothesis for R1, it suffices to find a legal color for v_t so that R2 and R3 hold. Note that we have

$$\mathrm{Forb}(v_t) \subseteq \{\phi(m(v_t)), \phi(m(v_t)) \pm 1, \phi(m(m(v_t))), \phi(f(m(v_t))), \phi(x)\},$$

where x is a possible already-colored sibling of v_t. So $|\mathrm{Forb}(v_t)| \leq 6$.

If $5 \leq s[w,t] < 9$, then $|C[w,t-1]| \geq 8$, so there exists some $j \in C[w,t-1] - \mathrm{Forb}(v_t)$. Let $\phi(v_t) = j$. Then R2 holds, while R3 is vacuous.

If $s[w,t] \geq 9$, then $|C[w,t-1]| \geq 12$. By inductive hypothesis for R1, $C[w,t-1]$ has at most 5 segments. By Proposition 8, $|C[w,t-1] - A(C[w,t-1])| \leq 4$. Since $|C[w,t-1]| \geq 12$, we have $|A(C[w,t-1]) \cap C[w,t-1]| \geq 8$. As $|\mathrm{Forb}(v_t)| \leq 6$, there exists some $j \in A(C[w,t-1]) \cap C[w,t-1]$ which is legal for v_t. Let $\phi(v_t) = j$. Then R2 and R3 hold, as $C[w,t-1] = C[w,t]$.

Case 2. $v_t \sim w$. Then it suffices to find a legal color for v_t such that R1 and R3 hold.

Assume $s[w,t] = 5$ or 6. Then $|C[w,t-1]| = 7$ or 8. By Lemma 9 (4), $|\mathrm{Forb}(v_t)| \leq 13$. Because $k \geq 15$, there exists some color $j \notin \mathrm{Forb}(v_t)$. By inductive hypothesis, $C[w,t-1]$ has at most 5 segments. If $C[w,t-1]$ contains less than 5 segments, then let $\phi(v_t) = j$. So, R1 holds, while R3 is vacuous. Suppose $C[w,t-1]$ contains exactly 5 segments (so 5 gaps). Since $|C[w,t-1]| \leq 8$ and $k \geq 15$, we conclude that there exists a gap with more than one element, so $|A(C[w,t-1]) - C[w,t-1]| \geq 6$. By Lemma 9 (4), there exists some $j \in A(C[w,t-1]) - \mathrm{Forb}(v_t)$. Let $\phi(v_t) = j$. Then R1 holds, while R3 is vacuous.

Assume $s[w,t] \geq 7$. Let $v_{t'} = m(v_t)$. By Lemma 7 (2, 4), $f(v_{t'}) = w$ and $s[w,t'] \geq 5$. By Lemma 9 (5), $|\mathrm{Forb}(v_t) - C[w,t-1]| \leq 4$. Moreover, by inductive hypothesis for R2, $\phi(v_{t'}) \in C[w,t'] \subseteq C[w,t-1]$. Therefore, we conclude that

$$\mathrm{Forb}(v_t) - C[w,t-1] \subseteq \{\phi(v_{t'}) \pm 1, \phi(m(v_{t'}))\}.$$

If $s[w,t] = 7, 8$, then $|C[w,t-1]| = 9, 10$, and hence $|\mathrm{Forb}(v_t)| \leq 13$. As $k \geq 15$, there exists some $j \notin \mathrm{Forb}(v_t)$. If $C[w,t-1]$ contains less than 5 segments, let $\phi(v_t) = j$. If $C[w,t-1]$ has exactly 5 segments (so 5 gaps), by Proposition 8 and $k \geq 15$ we have $|A(C[w,t-1]) - C[w,t-1]| \geq 5$. Thus, there exists some $j \in A(C[w,t-1]) - \mathrm{Forb}(v_t)$, as $|\mathrm{Forb}(v_t) - C[w,t-1]| \leq 3$. Let $\phi(v_t) = j$. Then R1 holds, while R3 is vacuous.

If $9 \leq s[w,t] \leq \Delta - 1$, then $s[w,t'] \geq 7$ and $|C[w,t-1]| \leq \Delta + 1$. By Lemma 7 (3), $w = f(m(v_{t'}))$. By inductive hypothesis for R2, $\{\phi(v_{t'}), \phi(m(v_{t'}))\} \subseteq C[w,t-1]$. So

$$\mathrm{Forb}(v_t) - C[w,t-1] \subseteq \{\phi(v_{t'}) \pm 1\}. \tag{3.2}$$

Because $|C[w,t-1]| \leq \Delta + 1$, $k = \Delta + 4$, and $\phi(v_{t'}) \in C[w,t-1]$, we conclude that $A(C[w,t-1]) - C[w,t-1] \not\subseteq \{\phi(v_{t'}) \pm 1\}$. Therefore, there exists a color $j \in A(C[w,t-1]) - \mathrm{Forb}(v_t)$. Let $\phi(v_t) = j$. Then R1 and R3 hold.

If $s[w,t] = \Delta \geq 11$, then $s[w,t'] \geq 9$. Similar to the above, (3.2) holds. Moreover, by R3, $\phi(v_{t'}) \in A(C[w,t']) \subset A(C[w,t-1])$, so one of $\phi(v_{t'}) \pm 1$ belongs to $C[w,t-1]$. Hence, $|\mathrm{Forb}(v_t) - C[w,t-1]| \leq 1$ and $|\mathrm{Forb}(v_t)| \leq \Delta + 3$ (because $|C[w,t-1]| \leq \Delta + 2$). Therefore, there exists some $j \in A(C[w,t-1]) - \mathrm{Forb}(v_t)$. Let $\phi(v_t) = j$. Then R1 and R3 hold. □

4. Proof of Theorem 1 and consequences.

Similar to the previous section, we prove Theorem 1 by giving a sequential coloring scheme based upon the ordering $\prec$. Since we have fewer colors, the sequential coloring scheme is more restrictive. We use the same notations as in the previous section. Let $k = \Delta + 3$ and assume ϕ is a partial k-$L_c(2,1)$-labeling for V_t, where $t \geq 3$. Let $w = f(v_t)$ and let β be the number of segments in $C[w,t]$. If w has degree Δ, then let u be its Δth neighbor; otherwise u does not exist and we simply ignore the parts involving u. Throughout the proof we call ϕ a *valid partial labeling* for V_t if R1–R4 in the following hold.

R1. $\beta \leq 6$; and if $w \not\prec m(u)$ or $s[w,t] \leq 9$, then $\beta \leq 5$.

R2. If $s[w,t] \geq 5$, then $\phi(v_t) \in C[w,t]$.

R3. If $s[w,t] \geq 11$, then $\phi(v_t) \in C[w,t] \cap A(C[w,t])$.

R4. Assume w has degree Δ (i.e., u exists). If $s[w,t] \geq 10$ and $v_t \prec m(u)$, then there exists some $j^* \in F(C[w,t])$, which is legal for $m(u)$. Moreover, if $w \not\prec m(u)$, then

$$j^* \in C[w,t] \cap F(C[w,t]) \ \text{ and } \ j^* \neq \phi(w).$$

We prove that for any $3 \leq t \leq n$, there is a valid partial labeling for V_t.

Initially: Let $\phi(v_1) = 0$, $\phi(v_2) = 2$, and $\phi(v_3) = 4$. Then R1 is true, while R2–R4 are vacuous.

Inductively: Assume that $t \geq 4$ and ϕ is a valid partial labeling for V_{t-1}.

If $s[w,t] \leq 4$, then $|C[w,t-1]| \leq 7$ and $|\mathrm{Forb}(v_t)| \leq 12$ (by Lemma 9 (4)). Let $\phi(v_t) = j$ for some $j \notin \mathrm{Forb}(v_t)$ (j exists because $k = \Delta + 3 \geq 18$). Then R1 follows by (3.1), while R2–R4 are vacuous.

Assume $s[w,t] \geq 5$. We consider two cases.

Case 1. $v_t \not\prec w$. Then $C[w,t-1] = C[w,t]$, regardless of what color is assigned to v_t. Hence, by inductive hypothesis for R1, it suffices to find a legal color for v_t such that R2–R4 hold.

By Lemma 9 (1, 2),

$$\mathrm{Forb}(v_t) \subseteq \{\phi(m(v_t)), \phi(m(v_t)) \pm 1, \phi(m(m(v_t))), \phi(f(m(v_t))), \phi(x)\},$$

where x is a possible already-colored sibling of v_t. (Note that $\phi(f(v_t))$ might be a forbidden color for v_t; however, as $f(v_t)$ is a parent of $m(v_t)$, so $\phi(f(v_t))$ is included in the set on the right-side above.) Hence, $|\mathrm{Forb}(v_t)| \leq 6$.

Assume $5 \leq s[w,t] \leq 9$. Then $|C[w,t-1]| = s[w,t] + 3 \geq 8$. Hence, $|C[w,t-1] - \mathrm{Forb}(v_t)| \geq 2$. Let $\phi(v_t) = j$ for some $j \in C[w,t-1] - \mathrm{Forb}(v_t)$. Then R2 holds, while R3 and R4 are vacuous.

Assume $s[w,t] \geq 10$. Let q be the largest index such that $q < t$ and $v_q \sim w$. Then $s[w,q] = s[w,t]$. By Lemma 7 (1), $w = f(v_q)$. If u exists and $v_t \prec m(u)$, then by inductive hypothesis (applied to $C[w,v_q]$) there exists some $j^* \in F(C[w,q]) = F(C[w,t-1])$ which is legal for $m(u)$. We need to find a legal color for v_t so that j^* is kept legal for $m(u)$.

If $s[w,t] = 10$, then $|C[w,t-1]| = 13$. As $|\mathrm{Forb}(v_t)| \leq 6$, there exists some $j \in C[w,t-1] - (\mathrm{Forb}(v_t) \cup \{j^*, j^* \pm 1\})$ (note that if u does not exist, we regard $\{j^*, j^* \pm 1\} = \emptyset$). Let $\phi(v_t) = j$. Then j^* is still legal for $m(u)$, and R2 and R4 hold (the "moreover" part of R4 follows by inductive hypothesis and $C[w,q] = C[w,t]$), while R3 is vacuous.

Assume $s[w,t] \geq 11$. Then $|C[w,t-1]| \geq 14$. By Lemma 7, $f(m(m(v_t))) = f(m(v_t)) = w$. By Lemma 9 (3), $|\mathrm{Forb}(v_t)| \leq 5$. By inductive hypothesis, $C[w,t-1]$ has at most 6 segments and, by definition, at most 5 of them are singletons. By Proposition 8, we have $|C[w,t-1] \cap A(C[w,t-1])| \geq 9$, implying that

$$|\,(C[w,t-1] \cap A(C[w,t-1])) - \{j^*, j^* \pm 1\}| \geq 6.$$

As $|\mathrm{Forb}(v_t)| \leq 5$, there exists some $j \in C[w,t-1] \cap A(C[w,t)) - \{j^*, j^* \pm 1\} - \mathrm{Forb}(v_t)$. Let $\phi(v_t) = j$ if $v_t \neq m(u)$, and $\phi(v_t) = j^*$ if $v_t = m(u)$. Then, R2–R4 hold, since $C[w,t] = C[w,t-1] = C[w,q]$.

Case 2. $v_t \sim w$. Then R2 holds, regardless of what color is assigned to v_t. So it suffices to find a legal color for v_t that satisfies R1, R3, and R4.

Assume $5 \leq s[w,t] \leq 9$. Then R3 and R4 are vacuous. Since $|C[w,t-1]| = |C[w,t]| - 1 = s[w,t] + 2 \leq 11$, by Lemma 9 (4), $|\mathrm{Forb}(v_t) - C[w,t-1]| \leq 5$. By inductive hypothesis, $C[w,t-1]$ has at most 5 segments. If $C[w,t-1]$ has less than 5 segments, let $\phi(v_t) = j$ for some $j \notin \mathrm{Forb}(v_t)$, so R1 holds. If $C[w,t-1]$ has exactly 5 segments (so 5 gaps), then since $|C[w,t-1]| \leq 11$ and $k \geq 18$, we conclude that there exists a gap with at least two elements, so $|A(C[w,t-1]) - C[w,t-1]| \geq 6$. By Lemma 9 (4), there exists some $j \in A(C[w,t-1]) - \mathrm{Forb}(v_t)$. Let $\phi(v_t) = j$. Then R1 holds.

Assume $s[w,t] \geq 10$. Let $v_{t'} = m(v_t)$ and $v_{t''} = m(v_{t'})$. Then $t', t'' < t$. By Lemma 7, $s[w,t'] \geq 8$, $s[w,t''] \geq 6$, and $w = f(v_{t'}) = f(v_{t''})$. By inductive hypothesis for R2, we have $\{\phi(v_{t'}), \phi(v_{t''})\} \subseteq C[w,t-1]$. Therefore, by Lemma 9 (1, 2, 3, 5), we have

$$\mathrm{Forb}(v_t) - C[w,t-1] \subseteq \{\phi(v_{t'}) \pm 1\}. \tag{4.1}$$

Lemma 10. *Let ϕ be a valid partial labeling for V_{t-1}. If $v_t \sim w$, $s[w,t] \geq 10$, and $|C[w,t-1]| \leq 15$, then $A(C[w,t)) - \mathrm{Forb}(v_t) \neq \emptyset$.*

Proof. Assume $|C[w,t-1]| \leq 15$. For any segment $\{j, j+1, \ldots, j'\}$ of $C[w,t-1]$, we have $\{j-1, j'+1\} \subseteq A(C[w,t-1]) - C[w,t-1]$ (note that $j-1 \neq j'+1$ since

$k \geq 18$). If $C[w,t-1]$ has more than one segment, then $|A(C[w,t-1]) - C[w,t)| \geq 3$. By (4.1), $A(C[w,t-1]) - \mathrm{Forb}(v_t) \neq \emptyset$.

Assume $C[w,t-1]$ has only one segment, say $C[w,t-1] = \{j, j+1, \ldots, j'\}$. Then $A(C[w,t-1]) - C[w,t-1] = \{j-1, j'+1\}$. As $\phi(v_{t'}) \in C[w,t-1]$ (see the line above 4.1), it follows that $\{j-1, j'+1\} \neq \{\phi(v_{t'}))\pm 1\}$. Therefore $A(C[w,t-1]) - \mathrm{Forb}(v_t) \neq \emptyset$. □

If u exists, then $u = v_b$, $m(u) = v_{b'}$, $m(v_{b'}) = v_{b''}$ for some $b'' < b' < b$, and $s[w,b] = \Delta \geq 15$. By Lemma 7, $s[w,b'] \geq 13$, $s[w,b''] \geq 11$, and $f(v_{b'}) = f(v_{b''}) = w$.

Assume $s[w,t] = 10$. Then R3 is vacuous, $s[w,t-1] \leq 9$, and $|C[w,t-1]| = 12$. We consider two subcases.

Subcase 2.A. $s[w,t] = 10$ and u does not exist or $m(u) \not\sim w$. By Lemma 10, there exists some $j \in A(C[w,t-1]) - \mathrm{Forb}(v_t)$.

Let $\phi(v_t) = j$. Then R1 holds by inductive hypothesis. If u does not exist, then R4 is vacuous, and we are done.

Assume u exists and $m(u) = v_{b'} \not\sim w$. It suffices to verify R4, that is, to find some $j^* \in F(C[w,t]) \cap C[w,t] - \{\phi(w)\}$ such that j^* is legal for $v_{b'}$. As $s[w, v_{b''}] \geq 11$, $v_t \prec v_{b''}$ (i.e., $v_{b''}$ has not been colored yet). Because $v_{b'} \not\sim w$, for j^* to be legal for $v_{b'}$ it suffices that $j^* \notin \{\phi(w), \phi(m(v_{b''}))\}$. Note that any segment of $C[w,t]$ has at most two ends, and all the colors in the segment except the ends are in $F(C[w,t])$. Because $C[w,t]$ has at most 5 segments and $|C[w,t]| = 13$, we have $|C[w,t] \cap F(C[w,t])| \geq 3$. Hence, there exists some $j^* \in C[w,t] \cap F(C[w,t]) - \{\phi(w), \phi(m(v_{b''}))\}$ such that j^* is legal for $m(u)$. So R4 is satisfied.

Subcase 2.B. $s[w,t] = 10$, u exists, and $m(u) \sim w$. In contrast to Subcase 2.A, we first fix j^* and then label v_t such that R1, R2, and R4 are satisfied.

Suppose $C[w,t-1]$ contains a singleton gap $\{i\}$. That is, $i \in F(C[w,t-1]) - C[w,t-1]$. Let $j^* = i$. We need to show there exists a legal color for v_t so that j^* is kept legal for $m(u)$. As $|C[w,t-1]| = 12$ and $\{j^*\}$ is a gap of $C[w,t-1]$ (so $C[w,t-1]$ contains at least two segments), by (4.1) and an argument similar to the proof of Lemma 10, there exists some $j \in A(C[w,t-1]) - \mathrm{Forb}(v_t) - \{j^*\}$. Let $\phi(v_t) = j$. Then R1 holds by inductive hypothesis. Because $s[w,b'] \geq 13$ and $s[w,t] = 10$, by Lemma 7 (5), we have $v_t \not\sim v_{b'}$. Hence j^* is legal for $v_{b'} = m(u)$, and R4 holds.

Now suppose that every gap in $C[w,t-1]$ contains at least two elements. Note that $C[w,t-1]$ contains at most 5 segments, as $s[w,t-1] \leq 9$. If there is only one gap, say $\{j, j+1, \ldots, j+i\}$, then since $|C[w,t-1]| = 12$ and $k \geq 18$, we have $i \geq 5$. It follows from (4.1) that $j+1$ or $j+i-1$ is legal for v_t. If there are at least two gaps, then since each gap contains at least two elements, it is easy to verify (again, using (4.1)) that there is a gap $\{j, j+1, \ldots, j+i\}$ such that $j+1$ or $j+i-1$ is legal for v_t. Accordingly, let $\phi(v_t) = j+1$ or $j+i-1$, and let $j^* = j$ or $j+i$, respectively. Then j^* satisfies R4. Moreover, $C[w,t]$ contains at most 6 segments. So R1 holds. This completes the proof for the case $s[w,t] = 10$.

Assume $s[w,t] = 11, 12$, so $|C[w,t-1]| = 13, 14$. Assume u exists and $m(u) \sim w$. By inductive hypothesis for R4, there exists some $j^* \in F(C[w,t-1])$ which is legal for $m(u)$. Then j^* must be a singleton gap of $C[w,t-1] = C[w,t-1]$, since $m(u) \sim w$. This implies that $C[w,t-1]$ contains at least two gaps. We claim

$$A(C[w,t-1]) - C[w,t-1] - \{\phi(v_{t'}) \pm 1, j^*\} \neq \emptyset. \tag{4.2}$$

If $C[w,t-1]$ has more than two gaps, then $|A(C[w,t-1]) - C[w,t-1]| \geq 4$. Therefore, (4.2) holds. If $C[w,t-1]$ has exactly two gaps, then $|A(C[w,t)) - C[w,t-1] - \{j^*\}| = 2$, since $k \geq 18$. Note that $A(C[w,t)) - C[w,t-1] - \{j^*\} \neq \{\phi(v_{t'}) \pm 1\}$, as $\phi(v_{t'}) \in C[w,t-1]$. So (4.2) holds.

Let $\phi(v_t) = j$ for some $j \in A(C[w,t-1]) - C[w,t-1] - \{\phi(v_{t'}) \pm 1, j^*\}$. This justifies R1 and R3 (since $\phi(v_t) \in C[w,t]$). Moreover, since $j \notin \{j^*, j^* \pm 1\}$ (as $\{j^* \pm 1\} \subseteq C[w,t-1]$), j^* is still a legal color for $m(u)$. So R4 holds.

Now assume that u does not exist, or u exists but $m(u) \not\sim w$. As $|C[w,t-1]| \leq 14$, by Lemma 10 there exists some $j \in A(C[w,t-1]) - \mathrm{Forb}(v_t)$. Let $\phi(v_t) = j$. Then R1 and R3 hold. If u does not exist, then R4 is vacuous. If u exists but $m(u) \not\sim w$, then by inductive hypothesis for R4, there exists some legal color j^* for $m(u)$ such that $\{j^*, j^* \pm 1\} \subseteq C[w,t-1]$. So $j \notin \{j^*, j^* \pm 1\}$, and j^* is still a legal color for $m(u)$. Hence, R4 holds.

Assume $13 \leq s[w,t] \leq \Delta - 1$. Then $v_t \neq u$. As $s[w,t'] \geq 11$, by inductive hypothesis and R3, $\phi(v_{t'}) \in A(C[w,t']) \cap C[w,t'] \subseteq A(C[w,t-1]) \cap C[w,t-1]$. Combining this with (4.1), we have $|\mathrm{Forb}(v_t) - C[w,t-1]| \leq 1$.

Suppose $v_t \prec m(u)$. If $s[w,t] \leq \Delta - 2$, then $|C[w,t-1]| \leq \Delta$. Because $k = \Delta + 3$, $|\mathrm{Forb}(v_t) - C[w,t-1]| \leq 1$, and $j^* \in F(C[w,t-1])$, we conclude that there exists some $j \in A(C[w,t-1]) - \mathrm{Forb}(v_t) - \{j^*\}$. Let $\phi(v_t) = j$. Then R1, R3 (since $\phi(v_t) \in C[w,t]$), and R4 hold.

If $s[w,t] = \Delta - 1$, then $m(u) \not\sim w$ (as $v_t \prec m(u)$), and $|C[w,t-1]| = \Delta + 1$. By inductive hypothesis, $\{j^*, j^* \pm 1\} \subseteq C[w,t-1]$. Therefore, there exists some $j \in A(C[w,t-1]) - \mathrm{Forb}(v_t)$, as $k = \Delta + 3$. Let $\phi(v_t) = j$. Then R1, R3, and R4 hold.

Suppose $v_t = m(u)$. Let $\phi(v_t) = j^*$. Then R1, R3, and R4 hold.

Suppose $m(u) \prec v_t$. Then $s[w,t] = \Delta - 1$ and $|C[w,t-1]| = \Delta + 1$. Because $|\mathrm{Forb}(v_t) - C[w,t-1]| \leq 1$, $|C[w,t)| = \Delta + 1$, and $k = \Delta + 3$, there exists some $j \in A(C[w,t)) - \mathrm{Forb}(v_t)$. Let $\phi(v_t) = j$. Then R1, R3, and R4 hold.

Assume $s[w,t] = \Delta \geq 15$. Then, $v_t = u$. By inductive hypothesis, $\phi(v_{t'}) = j^* \in F(C[w,t-1]) \cap C[w,t-1]$. By (4.1), we have $\mathrm{Forb}(v_t) = C[w,t-1]$. As $|C[w,t-1]| = \Delta + 2$ and $k = \Delta + 3$, we conclude that there is an attaching legal color for v_t.

In each of these cases, R2 holds by definition. This completes the proof of the validity of the coloring scheme. □

The following corollary follows from (1.1) and Theorems 1, 3, and 4.

Corollary 11.

$$\lambda(G) \leq \begin{cases} \Delta + 2 & \textit{if } G \textit{ is outerplanar with } \Delta(G) \geq 15; \\ \Delta + 3 & \textit{if } G \textit{ is outerplanar with } \Delta(G) \geq 11; \\ \Delta + 6 & \textit{if } G \textit{ is outerplanar}; \\ \Delta + 4 & \textit{if } G \textit{ is triangulated outerplanar.} \end{cases}$$

For outerplanar graphs with small maximum degrees, the equality of Theorem 1 does not always hold.

Theorem 12. *Let G_1, G_2, G_3, G_4 be the graphs as shown in Figure* 1 *(ignore the labels of vertices of G_4 for the moment). Then $\Delta(G_i) = i+1$, and $\lambda_c(G_i) = \Delta(G_i) + 4$.*

Proof. The proofs for G_1, G_2, G_3, and $\lambda_c(G_4) \leq \Delta(G_4)+4$ are straightforward. It is more complicated but routine to verify that $\lambda_c(G_4) > \Delta(G_4) + 3 = 8$. One method to accomplish this is (1) to prove that, by considering several cases, the labels (see Figure 1) for the "middle" induced subgraph H form a unique 8-$L_c(2,1)$-labeling for H; then (2) to show that this unique labeling cannot be extended to the vertices w and w'. We omit the details. □

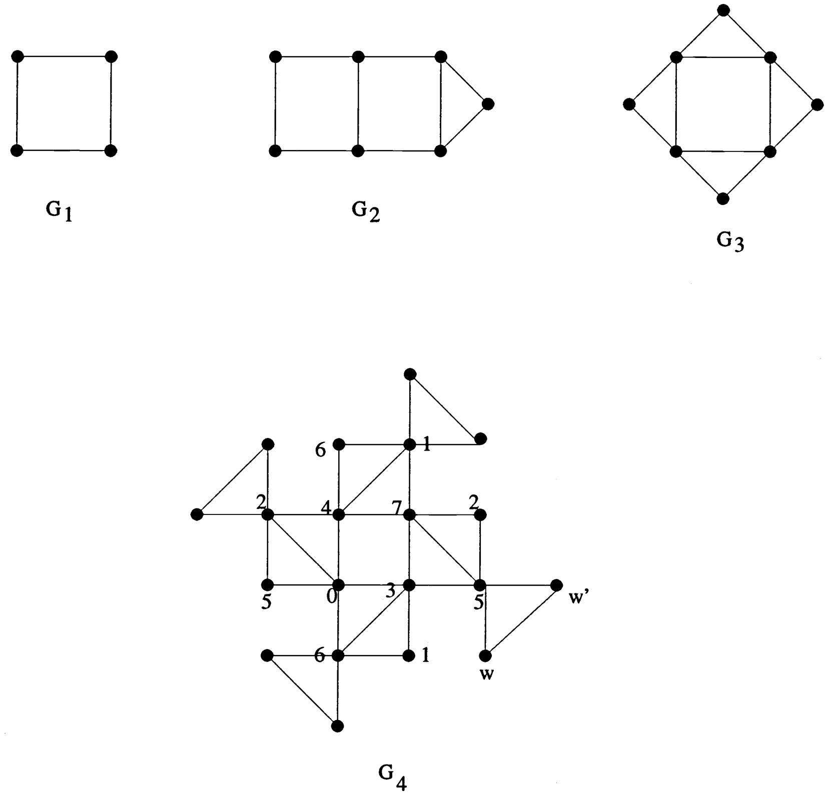


FIG. 1. *Graphs G_1, G_2, G_3, and G_4.*

Theorem 12 indicates that a condition like $\Delta(G) \geq 15$ is necessary for Theorem 1. Indeed, the authors of this article suspect that the condition might be replaced by $\Delta(G) \geq d$ for some $6 \leq d < 15$. Finding the smallest such integer d for Theorem 1 would be an interesting problem for further research.

Acknowledgment. The authors wish to thank the two anonymous referees for their helpful suggestions which resulted in a better presentation of this article.

REFERENCES

[1] H. L. Bodlaender, T. Kloks, R. B. Tan, and J. van Leeuwen, *Approximations for λ-Coloring of Graphs*, manuscript, 2001. An earlier version appeared in STACS 2000, Lecture Notes in Comput Sci. 1770, H. Reichel and S. Tichild, eds., Springer-Verlag, Berlin, Heidelberg, 2000, pp. 395–406.

[2] G. J. Chang and D. Kuo, *The $L(2,1)$-labeling problem on graphs*, SIAM J. Discrete Math., 9 (1996), pp. 309–316.

[3] J. R. Griggs and R. Yeh, *Labeling graphs with a condition at distance* 2, SIAM J. Discrete Math., 5 (1992), pp. 586–595.

[4] W. K. Hale, *Frequency assignment: Theory and applications*, Proc. IEEE, 68 (1980), pp. 1497–1514.

[5] J. van den Heuvel, R. A. Leese, and M. A. Shepherd, *Graph labelling and radio channel assignment*, J. Graph Theory, 29 (1998), pp. 263–283.

[6] D. Kráľ and R. Škrekovski, *A theorem about the channel assignment problem*, SIAM J. Discrete Math., 16 (2003), pp. 426–437.

[7] D. Liu, *Hamiltonicity and circular distance two labellings*, Discrete Math., 232 (2001), pp. 163–169.

[8] D. Liu, *Sizes of graphs with fixed orders and spans for circular-distance-two labeling*, Ars Combin., 67 (2003), pp. 125–139.

[9] D. Liu, *Circular Coloring for Graphs with Distance Two Condition*, manuscript, 2004.

[10] D. Liu and X. Zhu, *Circular distance two labeling and circular chromatic number*, Ars Combin., 69 (2003), pp. 177–183.

[11] B. Mohar, *Circular colorings of edge-weighted graphs*, J. Graph Theory, 43 (2003), pp. 107–116.

[12] M. Molloy and M. Salavatipour, *A bound on the chromatic number of the square of a planar graph*, J. Combin. Theory Ser. B, 94 (2005), pp. 189–213.

[13] D. Sakai, *Labeling chordal graphs: Distance two condition*, SIAM J. Discrete Math., 7 (1994), pp. 133–140.

SIAM J. DISCRETE MATH.
Vol. 19, No. 2, pp. 294–300

REPEATED ANGLES IN THREE AND FOUR DIMENSIONS*

ROEL APFELBAUM[†] AND MICHA SHARIR[‡]

Abstract. We show that the maximum number of occurrences of a given angle in a set of n points in $\mathbb{R}^3$ is $O(n^{7/3})$ and that a right angle can actually occur $\Omega(n^{7/3})$ times. We then show that the maximum number of occurrences of any angle different from $\pi/2$ in a set of n points in $\mathbb{R}^4$ is $O(n^{5/2}\beta(n))$, where $\beta(n) = 2^{O(\alpha(n)^2)}$ and $\alpha(n)$ is the inverse Ackermann function.

Key words. combinatorial geometry, geometric incidences, repeated angles

AMS subject classifications. 52C45, 68U05, 05D99, 51F99

DOI. 10.1137/S0895480104443941

1. Introduction.

In this paper we consider the following problem: Given a set P of n points in $\mathbb{R}^d$ and some fixed $0 < \alpha < \pi$, how many times can the angle α occur among triplets of points of P? That is, how many triplets $p, q, r \in P$ are there such that $\angle pqr = \alpha$? (We identify the triplet (p, q, r) with (r, q, p), and count them as only one angle.) The trivial upper bound is $O(n^3)$, which is the number of triplets, and a simple construction gives a lower bound of $\Omega(n^2)$ repeated angles.

In the plane, Pach and Sharir [4] have shown that the number of occurrences of a fixed angle among n points is $O(n^2 \log n)$ and that this lower bound can be achieved for every angle $\alpha = \arctan \frac{a\sqrt{m}}{b}$, where a, b, and m are positive integers.

In $\mathbb{R}^3$, the best known upper bound, $O(n^{8/3})$, is due to Conway et al. [3]; see also [1, section 6.2]. We improve this bound to $O(n^{7/3})$ and show that this bound is tight in case $\alpha = \pi/2$.

In $\mathbb{R}^4$, there is a construction of n points that determine $\Theta(n^3)$ *right angles* [1, section 6.2, problems 7 and 8], but for other angles $\alpha \neq \pi/2$, there is a subcubic bound of $O(n^{3-\frac{1}{25}})$, due to Purdy [6]; see [1, section 6.2]. We improve this bound by showing that the maximum number of repeated angles $\alpha \notin \{0, \frac{\pi}{2}, \pi\}$ in a set of n points in $\mathbb{R}^4$ is $O(n^{5/2}\beta(n))$, where $\beta(n) = 2^{O(\alpha(n)^2)}$ and $\alpha(n)$ is the inverse Ackermann function.

So far, the only lower bound in $\mathbb{R}^3$ and $\mathbb{R}^4$ that we have for $\alpha \notin \{0, \frac{\pi}{2}, \pi\}$ is the trivial bound $\Omega(n^2)$, and the planar bound $\Omega(n^2 \log n)$ for the above-mentioned special values of α.

As it turns out, the main difficulty in upper bounding the number of repeated angles lies in the possibility that the same angle instance is counted many times. Specifically, if $p \in P$ is incident to two rays that form an angle α, and if there are t points of P on each ray, then the same angle occurs t^2 times among t^2 triplets

*Received by the editors May 19, 2004; accepted for publication (in revised form) October 30, 2004; published electronically October 20, 2005. The work on this paper has been supported by a grant from the Israeli Academy of Sciences for a Center of Excellence in Geometric Computing at Tel Aviv University.

http://www.siam.org/journals/sidma/19-2/44394.html

[†]School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel (roel6@hotmail.com).

[‡]School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel, and Courant Institute of Mathematical Sciences, New York University, New York, NY 10012 (michas@post.tau.ac.il). The work of this author was also supported by a grant from the U.S.–Israel Binational Science Foundation, by NSF grant CCR-00-98246, and by the Hermann Minkowski–MINERVA Center for Geometry at Tel Aviv University.

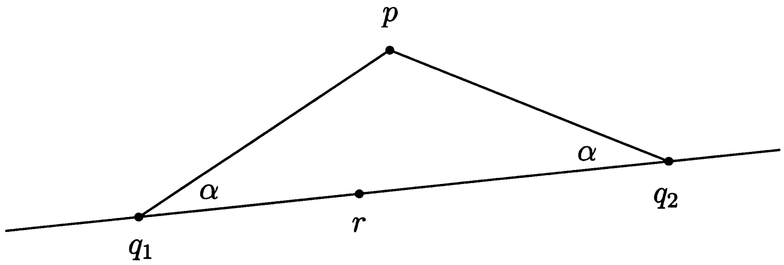


FIG. 1. *For a given point r and line ℓ, only two points $q \in \ell$ can be the apex of an angle $\angle pqr = \alpha$, with $p \in \ell$ too.*

of points of P. In particular, if $t = \Omega(n)$ we obtain the trivial lower bound $\Omega(n^2)$ mentioned above.

We overcome this difficulty by using the trade-off, due to Szemerédi and Trotter [7], between the number of rays containing many points and the number of points on each ray. The more points per ray, the fewer rays. More precisely, we have the following theorem.

THEOREM 1.1 (see Szemerédi and Trotter [7]). *Let P be a set of n points in the plane. Then the number of lines containing at least t points of P is $O(n^2/t^3 + n/t)$, and the number of incidences between these lines and the points of P is $O(n^2/t^2 + n)$.*

We note that the Szemerédi–Trotter theorem is stated for points and lines in the plane, but it can easily be extended to higher dimensions by projecting the given points and lines onto some generic plane. See [5, 7] for details.

2. Repeated angles in $\mathbb{R}^3$: Upper bound.

THEOREM 2.1. *Let $P \subset \mathbb{R}^3$ be a set of n points and let $0 < \alpha < \pi$ be fixed. Then the number of triplets $(p, q, r) \in P^3$ of distinct points satisfying $\angle pqr = \alpha$ is $O(n^{7/3})$.*

Proof. Denote the number of such triplets by $A(P)$. Let L be the set of lines spanned by P. Partition L into $m = \lceil \log n \rceil$ classes $L_1, L_2, \ldots, L_m$, so that the class L_i includes all the lines of L that contain at least 2^i and at most $2^{i+1} - 1$ points of P for $i = 1, \ldots, m$. We use in the proof the threshold value $k = \lceil \frac{1}{3} \log n \rceil$.

We say that an angle $\angle pqr$ is *supported* by the lines ℓ_1 and ℓ_2 if $\ell_1 = \overline{pq}$ and $\ell_2 = \overline{qr}$. For $1 \le j \le i \le m$, let $A_{i,j}$ denote the number of angles supported by one line from L_i and another line from L_j, that is,

$$A_{i,j} = \left| \left\{ (p, q, r) \in P^3 \;\middle|\; \angle pqr = \alpha,\ \overline{pq} \in L_i, \text{ and } \overline{qr} \in L_j \right\} \right|,$$

and let $A_i = \sum_{j=1}^{i} A_{i,j}$ (recall that we identify triplets (p, q, r) with their reverses (r, q, p)). We have $A(P) \le \sum_{i=1}^{m} A_i = \sum_{i=1}^{k} A_i + \sum_{i=k+1}^{m} A_i$. We shall bound separately the terms $A' = \sum_{i=1}^{k} A_i$ and $A'' = \sum_{i=k+1}^{m} A_i$.

To bound A'', we use the following easy but crucial observation. For each point $r \in P$ and a line $\ell \in L$, there are at most two points $q \in \ell \cap P$ such that r, q, and some third point, p, on ℓ form an angle α; see Figure 1. For each angle $\angle pqr = \alpha$ that is counted in A'', with $\overline{pq} \in L_i$, for some $i > k$, and $\overline{qr} \in L_j$, for some $j \le i$, we charge the triplet (p, q, r) to the pair (r, ℓ), where $\ell = \overline{pq}$. The preceding observation implies that if ℓ contains t points of P, where $2^i \le t < 2^{i+1}$, then the number of triplets that charge (r, ℓ) is at most $2t$, that is, at most twice the number of points of P on ℓ. This, in turn, implies that for a fixed $r \in P$ and for all lines ℓ containing 2^{k+1} points or more, the number of triplets that charge the pairs (r, ℓ) is at most twice the number of incidences between the points of P and these lines. Hence, using Theorem 1.1, we have

$$A'' = O(n(n^2 2^{-2k} + n)) = O(n^3 2^{-2k} + n^2) = O(n^{7/3}).$$

We next bound A'. Let $j \le i \le k$ be fixed. We bound the number of angles in $A_{i,j}$ in the following different way. Put $s = 2^i$ and $t = 2^j$. For a point $p \in P$, let ξ_p (resp., η_p) denote the number of rays emanating from p and contained in lines of L_i (resp., L_j). $\sum_{p \in P} \xi_p$ is twice the number of incidences between the points of P and the lines of L_i, since each such incidence, $(p, \ell) \in P \times L_i$, contributes exactly 2 to this sum by generating two opposite rays, that are counted in ξ_p. As noted above, applying Theorem 1.1 to the lines of L_i, each containing $\Theta(s)$ points, implies that

$$\sum_{p \in P} \xi_p = O\left(\frac{n^2}{s^2} + n\right) = O\left(\frac{n^2}{s^2}\right), \tag{2.1}$$

where the last equality follows from the fact that $s = O(n^{1/3})$. Similarly, we have

$$\sum_{p \in P} \eta_p = O\left(\frac{n^2}{t^2}\right). \tag{2.2}$$

Let σ_p denote the unit sphere centered at p. Map each ray emanating from p and contained in a line of L_i or L_j to its intersection point with σ_p. We thus obtain two sets C_p and D_p of ξ_p and η_p points, respectively, on the sphere σ_p, and we want to count the number of pairs in $C_p \times D_p$ at spherical distance exactly α. Each such pair corresponds to a pair of rays that emanate from p and subtend the angle α, so that one ray contains $O(s)$ points of P and the other contains $O(t)$ points. Hence each such pair generates $O(st)$ occurrences of the angle α among point triplets of P. The number of such pairs in $C_p \times D_p$ is equal to the number of incidences between ξ_p points and η_p congruent circles on σ_p and is thus bounded by $O((\xi_p \eta_p)^{2/3} + \xi_p + \eta_p)$ (see, e.g., [2]). Multiplying this by $O(st)$, and summing over all points p, we get

$$\begin{aligned} A_{i,j} &\le st \sum_{p \in P} O((\xi_p \eta_p)^{2/3} + \xi_p + \eta_p) \\ &= O\left(st \sum_{p \in P} (\xi_p \eta_p)^{2/3} + st \sum_{p \in P} \xi_p + st \sum_{p \in P} \eta_p\right). \end{aligned}$$

Using (2.1) and (2.2), the last two terms can be bounded by

$$O\left(st\left(\frac{n^2}{s^2} + \frac{n^2}{t^2}\right)\right) = O\left(\frac{n^2 t}{s} + \frac{n^2 s}{t}\right) = O\left(\frac{n^2 s}{t}\right),$$

since we have assumed that $s \ge t$. It remains to bound the first term. We observe that $\eta_p = O(n/t)$ for each $p \in P$, because all the rays emanating from p are pairwise disjoint (excluding the common point p). Combining this with Hölder's inequality and with the estimates (2.1) and (2.2), we thus have

$$\begin{aligned}\sum_{p\in P}(\xi_p\eta_p)^{2/3} &= (O(n/t))^{1/3}\sum_{p\in P}\xi_p^{2/3}\eta_p^{1/3}\\ &= O\left(n^{1/3}t^{-1/3}\left(\sum_{p\in P}\xi_p\right)^{2/3}\left(\sum_{p\in P}\eta_p\right)^{1/3}\right)\\ &= O\left(n^{1/3}t^{-1/3}\left(\frac{n^2}{s^2}\right)^{2/3}\left(\frac{n^2}{t^2}\right)^{1/3}\right)\\ &= O\left(n^{7/3}s^{-4/3}t^{-1}\right).\end{aligned}$$

This yields

$$A_{i,j} = O\left(n^{7/3}s^{-1/3}+\frac{n^2s}{t}\right) = O\left(n^{7/3}2^{-i/3}+n^22^{i-j}\right). \tag{2.3}$$

We then sum this bound over all $A_{i,j}$'s that contribute to A', to obtain

$$\begin{aligned}A' &= \sum_{i=1}^{k}\sum_{j=1}^{i}A_{i,j}\\ &= \sum_{i=1}^{k}\sum_{j=1}^{i}O\left(n^{7/3}2^{-i/3}+n^22^{i-j}\right)\\ &= O\left(n^{7/3}\sum_{i=1}^{k}\sum_{j=1}^{i}2^{-i/3}+n^2\sum_{i=1}^{k}\sum_{j=1}^{i}2^{i-j}\right)\\ &= O\left(n^{7/3}\sum_{i=1}^{k}i2^{-i/3}+n^2\sum_{i=1}^{k}2^{i}\right)\\ &= O\left(n^{7/3}+n^22^k\right)\\ &= O\left(n^{7/3}\right).\end{aligned}$$

Hence the number of repeated angles in P is at most $A'+A''=O(n^{7/3})$. □

3. Repeated angles in $\mathbb{R}^3$: Lower bound. In this section we show that the set P of vertices of the $n^{1/3}\times n^{1/3}\times n^{1/3}$ cubic lattice section determine $\Omega(n^{7/3})$ right angles. The proof outline is as follows. The points of P determine $O(n^{2/3})$ distinct distances. Hence, if we take all the spheres centered at points of P and containing at least one point of P, we get at most $O(n^{5/3})$ spheres. For simplicity we consider only the spheres fully contained in the bounding cube of P. On each sphere we obtain many right angles as follows. Take a pair of antipodal points $p,r\in P$ and another point $q\in P$ on the sphere. Then $\angle pqr=\pi/2$. On average there are $m=\Omega(n^{1/3})$ points on the sphere. There are $m/2$ choices of an (unordered) antipodal pair (p,r), and $m-2$ choices of a third point q, yielding about $m^2/2=\Omega(n^{2/3})$ right angles per sphere on average. Multiplying this bound by the number of spheres, $O(n^{5/3})$, we obtain that P determines $\Omega(n^{7/3})$ right angles.

In more detail, we have the next theorem.

Theorem 3.1. *Let $P = \{1, \dots, \lfloor n^{1/3} \rfloor\}^3$. Then the number of triplets $(p, q, r) \in P^3$ such that $\angle pqr = \pi/2$ is $\Omega(n^{7/3})$.*

Proof. For simplicity we assume that n is a cubic integer and a multiple of 5, so that all the quantities that appear in the proof are integers. This assumption does not change the order of magnitude of the lower bound.

Let $Q = \{\frac{2}{5}n^{1/3} + 1, \dots, \frac{3}{5}n^{1/3}\}^3$ be the middle $\frac{1}{5}n^{1/3} \times \frac{1}{5}n^{1/3} \times \frac{1}{5}n^{1/3}$ portion of P. We have $|Q| = \frac{n}{125} = \Theta(n)$. For each pair of points in Q, the square of the distance between them is an integer of magnitude at most $\frac{3}{25}n^{2/3}$. Hence there are at most $\frac{3}{25}n^{2/3} = O(n^{2/3})$ distinct distances between the points of Q. For every point $o \in Q$ we take the spheres centered at o and containing at least one point $p \in Q$. There are $O(n^{2/3})$ such spheres. We repeat this for all points of Q and let S denote the resulting set of spheres. We have $|S| = O(n^{5/3})$. The choice of Q guarantees that, for every point $p \in P$ on a sphere $\sigma \in S$, the point on σ antipodal to p is also in P.

For each $\sigma \in S$, let $m_\sigma = |P \cap \sigma|$ denote the number of lattice points on σ. We observe that $\sum_{\sigma \in S} m_\sigma \geq 2\binom{|Q|}{2} = \Omega(n^2)$, since in this sum we count every pair $p, p' \in Q$ exactly twice—once with p at the center of the sphere and p' on the sphere itself and once the other way around. Similarly, $\sum_{\sigma \in S} m_\sigma \leq |Q| \cdot |P| = O(n^2)$, so this sum is $\Theta(n^2)$. Let $\sigma \in S$ be one of the spheres and let $p, q, r \in \sigma \cap P$ be three distinct points such that p and r are antipodal points of σ. Then $\angle pqr = \pi/2$. There are $m_\sigma/2$ choices of an antipodal pair $p, r \in \sigma \cap P$ and $m_\sigma - 2$ choices of a third point q, yielding $m_\sigma(m_\sigma - 2)/2$ right angles on σ. The lower bound on the number of right angles in P is obtained by summing over all the spheres of S. Note that each pair of points can be antipodal on at most one sphere; hence every angle is counted only once. This gives a lower bound of

$$\frac{1}{2}\sum_{\sigma \in S} m_\sigma(m_\sigma - 2) \geq \frac{1}{2\,|S|}\left(\sum_{\sigma \in S} m_\sigma\right)^2 - \sum_{\sigma \in S} m_\sigma = \frac{1}{2\,|S|}\Theta(n^4) - \Theta(n^2),$$

where we have used the Cauchy–Schwarz inequality. Substituting $|S| = O(n^{5/3})$ in the inequality gives $\Omega(n^{7/3})$ right angles determined by the points of P. □

Remark. It is an interesting open problem whether the same lower bound also holds for other angles $\neq \pi/2$.

4. Repeated angles in $\mathbb{R}^4$.

Recall that there is a construction of n points in $\mathbb{R}^4$ that determine $\Theta(n^3)$ right angles, but for other angles $\alpha \neq \pi/2$, there is a subcubic upper bound of $O(n^{3-\frac{1}{25}})$, due to Purdy [6]; see [1, section 6.2]. In this section we improve this upper bound and derive the following result.

Theorem 4.1. *Let $P \subset \mathbb{R}^4$ be a set of n points and let $\alpha \notin \{0, \pi/2, \pi\}$ be fixed. Then the number of triplets $(p, q, r) \in P^3$ of distinct points satisfying $\angle pqr = \alpha$ is $O(n^{5/2}\beta(n))$, where $\beta(n) = 2^{c\alpha^2(n)}$ for some constant $c > 0$, and where $\alpha(n)$ is the extremely slowly growing inverse Ackermann function.*

Proof. The machinery of section 2 can be easily extended to four dimensions as follows. We use the same partition of the set L of lines spanned by P into $\lceil \log n \rceil$ classes, where the ith class consists of all lines that contain at least 2^i and at most $2^{i+1} - 1$ points of P. The values $A(P)$, $A_{i,j}$, and A_i are defined as in the three-dimensional case. Unlike the case of $\mathbb{R}^3$, we use the threshold value $k = \lceil \frac{1}{4}\log n \rceil$ and obtain $A(P) = A' + A''$, where $A' = \sum_{i \leq k} A_i$ and $A'' = \sum_{i > k} A_i$. Bounding A'' proceeds exactly as before and yields $A'' = O(n^3 2^{-2k}) = O(n^{5/2})$.

For A', we bound each of the $A_{i,j}$ terms separately. We set $s = 2^i$ and $t = 2^j$. As before, for each $p \in P$, we take the unit 3-sphere σ_p centered at p, intersect the rays emanating from p and contained in the lines of L_i and L_j with σ_p, and reduce the problem to that of counting repeated distances on the sphere σ_p, all equal to the spherical distance α. Each such incidence defines a pair of rays at angle α emanating from p, which contribute $O(st)$ angles to our count. The number of repeated distances on σ_p is equal to the number of incidences between ξ_p points and η_p congruent copies of the sphere $\mathbb{S}^2$ scaled according to the spherical distance α. If $\alpha \neq \pi/2$, then these copies of $\mathbb{S}^2$ are not great spheres on σ_p, which easily implies that three distinct copies intersect in at most two points (and not in a common circle, as may happen if they are great spheres).

We can then apply the analysis of [2, section 6] for the number of incidences between points and unit spheres in $\mathbb{R}^3$. Since our spheres lie on a 3-sphere rather than in Euclidean 3-space, the analysis of [2] requires some easy modifications. For example, we can project σ_p onto $\mathbb{R}^3$, using stereographic projection. The η_p 2-spheres on σ_p are then mapped to 2-spheres in $\mathbb{R}^3$, not necessarily of equal radius. Nevertheless, the analysis in [2] carries over to this situation. The two main properties that the analysis uses are that the incidence graph between the projected points and spheres does not contain $K_{3,3}$ and that the size of the vertical decomposition of an arrangement of r projected spheres is $O(r^3\beta(r))$, and, as is easily verified, both properties hold for the projected spheres and points.

We conclude that the number of incidences between ξ_p points and η_p 2-spheres on σ_p is $O((\xi_p\eta_p)^{3/4}\beta(\xi_p,\eta_p)+\xi_p+\eta_p)$, where $\beta(m,n) = 2^{c'\alpha^2(m^3/n)}$ for some constant $c' > 0$ independent of m and n, and where $\alpha(\cdot)$ is the inverse Ackermann function. Put $\beta(n) = \beta(n,1) = 2^{c'\alpha^2(n^3)}$. Since $\alpha(\cdot)$ is very slowly growing, we have $\alpha(n^3) = O(\alpha(n))$ and consequently $\beta(n) \leq 2^{c\alpha^2(n)}$ for an appropriate constant $c > 0$ depending only on c'. Note that $\beta(m,n)$ is ascending in m and descending in n, hence $\beta(\xi_p,\eta_p) \leq \beta(n,1)$ (unless $\eta_p = 0$, but in that case we trivially have 0 angle instances from $A_{i,j}$ at the apex p). Plugging this bound into an appropriately modified variant of the analysis of section 2 gives

$$\begin{aligned}
A_{i,j} &= O\left(st\sum_{p\in P}(\xi_p\eta_p)^{3/4}\beta(\xi_p,\eta_p) + st\sum_{p\in P}\xi_p + st\sum_{p\in P}\eta_p\right)\\
&= O\left(st\sum_{p\in P}(\xi_p\eta_p)^{3/4}\beta(n,1) + st\sum_{p\in P}\xi_p + st\sum_{p\in P}\eta_p\right)\\
&= O\left(st\beta(n)\sum_{p\in P}(\xi_p\eta_p)^{3/4} + \frac{n^2t}{s} + \frac{n^2s}{t}\right).
\end{aligned}$$

As above, we have

$$\begin{aligned}
\sum_{p\in P}(\xi_p\eta_p)^{3/4} &= (O(n/t))^{1/2}\sum_{p\in P}\xi_p^{3/4}\eta_p^{1/4}\\
&= O\left(n^{1/2}t^{-1/2}\left(\sum_{p\in P}\xi_p\right)^{3/4}\left(\sum_{p\in P}\eta_p\right)^{1/4}\right)
\end{aligned}$$

$$= O\left(n^{1/2}t^{-1/2}\left(\frac{n^2}{s^2}\right)^{3/4}\left(\frac{n^2}{t^2}\right)^{1/4}\right)$$
$$= O\left(n^{5/2}s^{-3/2}t^{-1}\right)$$

and thus

$$A_{i,j} = O\left(n^{5/2}\beta(n)2^{-i/2} + n^2 2^{i-j}\right). \tag{4.1}$$

Finally we sum over all the relevant $A_{i,j}$'s to obtain

$$A' = O\left(n^{5/2}\beta(n)\sum_{i=1}^{k}\sum_{j=1}^{i}2^{-i/2} + n^2\sum_{i=1}^{k}\sum_{j=1}^{i}2^{i-j}\right)$$
$$= O\left(n^{5/2}\beta(n) + n^2 2^k\right)$$
$$= O\left(n^{5/2}\beta(n)\right).$$

Hence the number of repeated angles in $\mathbb{R}^4$ is $O(n^{5/2}\beta(n))$. □

Remarks. The lower bound construction for $\mathbb{R}^3$ can be easily extended to $\mathbb{R}^4$ to yield a lower bound of $\Omega(n^{5/2})$ right angles, but this bound is very weak, since, as mentioned, right angles can be repeated $\Theta(n^3)$ times in $\mathbb{R}^4$. An interesting open problem is to match the upper bound of Theorem 4.1 by a lower bound close to $\Omega(n^{5/2})$. As mentioned in the introduction, the only lower bounds that we have so far (for $\alpha \neq \pi/2$) are $\Omega(n^2)$ and $\Omega(n^2 \log n)$ for the special values of α used in [4].

REFERENCES

[1] P. Brass, W. Moser, and J. Pach, *Research Problems in Discrete Geometry*, Springer-Verlag, New York, to appear.

[2] K. L. Clarkson, H. Edelsbrunner, L. Guibas, M. Sharir, and E. Welzl, *Combinatorial complexity bounds for arrangements of curves and spheres*, Discrete Comput. Geom., 5 (1990), pp. 99–160.

[3] J. H. Conway, H. T. Croft, P. Erdős, and M. J. T. Guy, *On the distribution of values of angles determined by coplanar points*, J. London Math. Soc. (2), 19 (1979), pp. 137–143.

[4] J. Pach and M. Sharir, *Repeated angles in the plane and related problems*, J. Combin. Theory Ser. A, 59 (1992), pp. 12–22.

[5] J. Pach and M. Sharir, *Geometric incidences*, in Towards a Theory of Geometric Graphs, J. Pach, ed., Contemp. Math. 342, AMS, Providence, RI, 2004, pp. 185–223.

[6] G. Purdy, *Repeated angles in E_4*, Discrete Comput. Geom., 3 (1988), pp. 73–75.

[7] E. Szemerédi and W. Trotter, *Extremal problems in discrete geometry*, Combinatorica, 3 (1983), pp. 381–392.

SIAM J. Discrete Math.
Vol. 19, No. 2, pp. 301–303

CONSTRUCTIVE BOUNDS ON ORDERED FACTORIZATIONS[*]

DON COPPERSMITH[†] AND MOSHE LEWENSTEIN[‡]

Abstract. The number of ways to factor a natural number into an ordered product of integers, each factor greater than one, is called the *ordered factorization of* n and is denoted $H(n)$. We show upper and lower bounds on $H(n)$ with explicit constructions.

Key words. factorizations, Riemann zeta function

AMS subject classifications. 05A17, 05A15, 05A05

DOI. 10.1137/S0895480104445861

1. Introduction.

For $n \in \mathbb{Z}^+$, let $H(n)$ denote the number of *ordered factorizations* of n, by which we mean expressions of n as the product of integers $r_i \geq 2$ where the order of factors is essential. Equivalently, $H(n)$ is the number of tuples $(r_1, r_2, \ldots, r_k)$ with $r_i \geq 2$ and $\prod r_i = n$, without restrictions on k. $H(1) = 1$ by convention, the only factorization being () with $k = 0$. $H(20) = 8$, the factorizations being (20), (10,2), (5,4), (5,2,2), (4,5), (2,10), (2,5,2), (2,2,5). Newberg and Naor [3] use $H(n)$ as a lower bound for an application in computational biology.

Define

$$\rho = \zeta^{-1}(2) \approx 1.72864724,$$

where ζ is the Riemann zeta function, so that

$$\sum_{n=1}^{\infty} \frac{1}{n^\rho} = 2$$

and, more usefully,

$$\sum_{n=2}^{\infty} \frac{1}{n^\rho} = 1.$$

Hille [2] showed the existence of a constant c such that $H(n) \leq cn^\rho$; Chor, Lemke, and Mador [1] improved this to $c = 1$:

$$H(n) \leq n^\rho. \tag{1}$$

Hille also gave an existential lower bound: for all $\epsilon > 0$,

$$\limsup_n \frac{H(n)}{n^{\rho-\epsilon}} = \infty. \tag{2}$$

Newberg and Naor show an explicit construction lower bounding $H(n)$ with $n \log^c n$ for some c. Chor, Lemke, and Mador gave explicit constructions for certain values of ϵ.

In this note we give simplified proofs of both upper and lower bounds.

[*] Received by the editors September 10, 2004; accepted for publication (in revised form) October 27, 2004; published electronically October 20, 2005.
http://www.siam.org/journals/sidma/19-2/44586.html

[†] IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 (dcopper@us.ibm.com).
[‡] Bar Ilan University, Ramat Gan 52900, Israel (moshe@cs.biu.ac.il).

2. Upper bound. The upper bound $H(n) \le n^\rho$ is proved by induction on n. The base case $n = 1$ is satisfied. Suppose the result is true for all $n' < n$. We count the ordered factorizations of n according to their first element r_1, which is a factor of n larger than 1. The remainder $(r_2, \dots, r_k)$ is an ordered factorization of n/r_1. So we have

$$H(n) = \sum_{d|n, d>1} H(n/d).$$

By induction,

$$H(n/d) \le (n/d)^\rho,$$

so that

$$\begin{aligned} H(n) &= \sum_{d|n,d>1} H(n/d) \le \sum_{d|n,d>1} \frac{n^\rho}{d^\rho} < n^\rho \sum_{d>1} \frac{1}{d^\rho} \\ &= n^\rho(\zeta(\rho) - 1) = n^\rho(2-1) = n^\rho, \end{aligned}$$

completing the induction. In fact, we see that the inequality is strict for $n > 1$.

3. Lower bound. For $\alpha = \rho - \epsilon$ we will give a family of integers n for which

$$\limsup_n H(n)/n^\alpha = \infty.$$

Because $\zeta(t)$ is strictly monotone decreasing in t, we know

$$\zeta(\alpha) = \sum_1^\infty \frac{1}{n^\alpha} > 2.$$

There is a finite integer b for which already

$$\sum_1^b \frac{1}{n^\alpha} > 2.$$

Use monotonicity again to claim there is γ with $\alpha < \gamma < \rho$ satisfying

$$\sum_1^b \frac{1}{n^\gamma} = 2$$

or, more usefully,

$$\sum_2^b \frac{1}{n^\gamma} = 1.$$

Fix such α, b, γ.

Now select a large integer t. For $k = 2, 3, \dots, b$, we define

$$c_k = \lfloor t/k^\gamma \rfloor.$$

Set $u = \sum c_k$ so that $0 \le t - u \le b - 2$. Define

$$n = \prod_{k=2}^b k^{c_k}.$$

We will compare $H(n)$ to n^α. Among the ordered factorizations counted by $H(n)$ are the orderings of (c_2 copies of $2, \ldots, c_b$ copies of b). The number of such orderings is given by the multinomial coefficient

$$v(n) = \frac{u!}{\prod_{k=2}^{b} c_k!}.$$

From Stirling's approximation,

$$v(n) = \prod_k \left(\frac{u}{c_k}\right)^{c_k} \times \sqrt{\frac{2\pi u}{\prod(2\pi c_k)}} \times [1 + o(1)],$$

where the $o(1)$ term goes to 0 with increasing c_k and hence with increasing n.

To estimate the first product, recall $c_k \leq t/k^\gamma$, so that

$$\prod_k \left(\frac{u}{c_k}\right)^{c_k} \geq \prod_k \left(\frac{uk^\gamma}{t}\right)^{c_k} = (u/t)^u \left(\prod_k k^{c_k}\right)^\gamma.$$

We have $(u/t)^u \geq e^{-(t-u)} \geq e^{-b+2}$, while the other factor is simply n^γ. So our first product is at least $e^{-b+2} n^\gamma$.

The second product is

$$\sqrt{\frac{2\pi u}{\prod(2\pi c_k)}}.$$

Notice that $\log n = \sum c_k \log k$, which implies that $\log n < \sum(c_k \log b)$. Hence, $u = \sum c_k > (\log n/\log b)$. On the other hand, for any $k, c_k < \sum(c_k \log k) = \log n$. Therefore, for some constant c_b we can lower bound the second product as follows:

$$\sqrt{\frac{2\pi u}{\prod(2\pi c_k)}} > c_b(\log n)^{-(b-2)/2}.$$

Summarizing,

$$H(n) \geq v(n) \geq n^\gamma (\log n)^{-(b-2)/2} c_b(1 + o(1)).$$

Since $\gamma > \alpha$, we have

$$\limsup_n H(n)/n^\alpha = \infty,$$

as required.

REFERENCES

[1] B. Chor, P. Lemke, and Z. Mador, *On the number of ordered factorizations of natural numbers*, Discrete Math., 214 (2000), pp. 123–133.

[2] E. Hille, *A problem in factorisation numerorum*, Acta Arithmetica, 2 (1936), pp. 134–144.

[3] L. A. Newberg and D. Naor, *A lower bound on the number of solutions to the probed partial digestion problem*, Adv. Appl. Math., 14 (1993), pp. 172–183.

SIAM J. Discrete Math.
Vol. 19, No. 2, pp. 304–320

A STRONGLY POLYNOMIAL CUT CANCELING ALGORITHM FOR MINIMUM COST SUBMODULAR FLOW[*]

SATORU IWATA[†], S. THOMAS MCCORMICK[‡], AND MAIKO SHIGENO[§]

Abstract. This paper presents a new strongly polynomial cut canceling algorithm for minimum cost submodular flow. The algorithm is a generalization of our similar cut canceling algorithm for ordinary min-cost flow. The algorithm scales a relaxed optimality parameter and creates a second, inner relaxation that is a kind of submodular max flow problem. The outer relaxation uses a novel technique for relaxing the submodular constraints that allows our previous proof techniques to work. The algorithm uses the min cuts from the max flow subproblem as the relaxed most positive cuts it chooses to cancel. We show that this algorithm needs to cancel only $O(n^3)$ cuts per scaling phase, where n is the number of nodes. Furthermore, we show how to slightly modify this algorithm to get a strongly polynomial running time. Finally, we briefly show how to extend this algorithm to the separable convex cost case and that the same technique can be used to construct a polynomial time maximum mean cut canceling algorithm for submodular flow.

Key words. combinatorial optimization, strongly polynomial time algorithm, submodular function, network flow

AMS subject classifications. 90C27, 90C35

DOI. 10.1137/S0895480199361533

1. Introduction.

A fundamental problem in combinatorial optimization is min-cost network flow (MCF). It can be modeled as a linear program with guaranteed integral optimal solutions (with integral data), and many polynomial and strongly polynomial algorithms for it exist (see Ahuja, Magnanti, and Orlin [1] for background). Researchers in mathematical programming have developed a series of extensions of MCF having integral optimal solutions (see Schrijver [40]).

Often the proofs of integrality are existential rather than algorithmic. We have long been interested in finding a *generic* (strongly) polynomial algorithm for such problems, i.e., one that is easily extended from the MCF case to more general cases. Finding such an algorithm would allow us to better understand which features of MCF algorithms depend on special structure and which ones are more general. We hope that it would also allow researchers to be able to find algorithms for more general problems more quickly than in the past.

Natural classes of generic algorithm to consider are the classes of (primal) cycle canceling algorithms and (dual) cut canceling algorithms. These are natural in the sense that they take improving steps coming from the linear algebra of the constraints, and their sense of "improvement" depends only on the local effect on the objective

[*] Received by the editors September 17, 1999; accepted for publication (in revised form) November 12, 2004; published electronically October 20, 2005. An extended abstract of a preliminary version of this paper appeared in *Integer Programming and Combinatorial Optimization*, Springer, Berlin, 1999, pp. 259–272.
http://www.siam.org/journals/sidma/19-2/36153.html

[†] Department of Mathematical Informatics, Graduate School of Information Science and Technology, University of Tokyo, Tokyo 113-8656, Japan (iwata@sr3.t.u-tokyo.ac.jp).

[‡] The Sauder School of Business, University of British Columbia, Vancouver, BC V6T 1Z2, Canada (Tom.McCormick@sauder.ubc.ca). This author's research was supported by an NSERC operating grant; part of this research was done during visits to RIMS at Kyoto University and SORIE at Cornell University.

[§] Institute of Policy and Planning Sciences, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan (maiko@shako.sk.tsukuba.ac.jp).

value. It is (in theory) easy to figure out what these objects should look like for more general models than MCF.

A natural first step in applying such a research program is to consider the case of *submodular flow* (SF), formally defined in section 2. This problem very much resembles ordinary MCF, except that the usual conservation constraints have been relaxed into constraints on the total violation of conservation on any node subset. SF was shown to enjoy integral optimal solutions by Edmonds and Giles [4]. Early algorithmic contributions came from [3, 9, 10, 13]. Our algorithm uses many ideas from these papers. The first strongly polynomial algorithm for SF is due to Frank and Tardos [11], who generalized the strongly polynomial MCF algorithm by Tardos [43] to a fairly wide class of combinatorial optimization problems. A more direct generalization of the Tardos algorithm was presented by Fujishige, Röck, and Zimmermann [18] with the aid of the tree-projection method by Fujishige [15].

Unfortunately it has been surprisingly difficult to extend known MCF cycle and cut canceling algorithms to SF. One of the most attractive MCF canceling algorithms is the min mean cycle canceling algorithm of Goldberg and Tarjan [20] (and its dual, the max mean cut canceling algorithm of Ervolina and McCormick [5]). Cui and Fujishige [2] were able to show finiteness of this algorithm for SF, but no polynomial bound. McCormick, Ervolina, and Zhou [35] show that it is unlikely that a straightforward generalization of either min mean cycle or max mean cut canceling is polynomial for SF. In 1999 Wallacher and Zimmermann [44] devised a weakly polynomial cycle canceling algorithm for SF based on the min ratio approach by Zimmermann [45]. This algorithm is provably not strongly polynomial, even for networks [36].

Another possibility, developed and analyzed in [42], is to cancel cycles or cuts which maximize the violation of complementary slackness with respect to a relaxed optimality parameter, the *relaxed min/max* canceling algorithms. These algorithms allow for a simpler computation of the cycle or cut to cancel, but otherwise enjoy the relatively simple analysis of min mean cycle/max mean cut canceling.

The same authors found a way to generalize relaxed min cycle canceling to SF [28], which led to nearly the fastest weakly polynomial running time for SF. This algorithm leads to the same strongly polynomial bound as the fastest known one by Fujishige, Röck, and Zimmermann [18], and it can also be extended to SF with separable convex costs. The same ideas can be further extended to solve separable convex cost problems over any totally unimodular system [29].

However, in some circumstances cut canceling appears to be more generalizable than cycle canceling. One recent example of this is the problem of submodular function minimization (SFM; see [32] for a survey). One of the two basic SFM algorithms, by Iwata, Fleischer, and Fujishige [25], is a direct descendent of the cut canceling algorithm in the present paper, whereas no cycle canceling SFM algorithm is known. Hence it is important to also generalize cut canceling MCF algorithms to SF.

This paper extends the relaxed most positive cut (MPC) canceling algorithm for MCF of [42] in a nontrivial way to SF. Section 3 develops the concepts of dual approximate optimality we will need, and section 4 develops the main weakly polynomial algorithm. The algorithm runs in $O(n^6 h \log(nU))$ time, where n is the number of vertices, h is the time for computing an exchange capacity, and U is the maximum absolute value of capacities. The next three sections generalize this algorithm in three directions: to a strongly polynomial algorithm (section 5), to the case of separable convex costs (section 6), and to a max mean cut variant (section 7). The total running time bound of the strongly polynomial algorithm is $O(n^8 h \log n)$. Finally, in section 8, we describe how to apply our algorithms to the SF problem with crossing submodular functions.

The only other polynomial cut canceling algorithm we know for SF is the most helpful total cut canceling algorithm of [33]. However, that algorithm is provably not strongly polynomial [6], and it needs an oracle to compute exchange capacities for a derived submodular function, which appears to be difficult to derive from an oracle for the original submodular function. By contrast, the present algorithm can compute exchange capacities in derived networks using only an oracle for the original function, and is the first dual strongly polynomial algorithm for SF that we know of. Our algorithm also appears to be the first strongly polynomial algorithm (primal or dual) for SF that avoids scaling or rounding the data. The original version of this paper [26] developed the first strongly polynomial cut canceling algorithm for SF. Subsequently, Fleischer, Iwata, and McCormick (FIM) [7] derived a new strongly polynomial SF algorithm which improves the running time of the capacity scaling algorithm for SF in [24] by incorporating the cut canceling technique developed in the present paper. The FIM algorithm improves the cut canceling SF algorithm of this paper by updating dual prices using shortest path distances coming from a modified form of Dijkstra's algorithm to effectively cancel many cuts at once. The FIM algorithm improves both the weakly and the strongly polynomial bounds of this paper by a factor of n^2. The strongly polynomial bound matches the current best-known bound, and the weakly polynomial bound is better than any previously known algorithm for the SF problem when costs are large but capacities are not (again showing that cut canceling algorithms are sometimes superior). The FIM SF algorithm then led to the IFF SFM algorithm [25], which solved one of the most important and long-standing open problems in combinatorial optimization. This development validates our claim that this research program can lead to quicker algorithm development, as the IFF algorithm followed the FIM algorithm relatively quickly, despite SFM seeming to be quite different from SF.

2. Submodular flow. An instance of submodular flow looks much like an instance of MCF. We are given a directed graph $G = (N, A)$ with node set N of cardinality n and arc set A of cardinality m. We are also given *lower* and *upper bounds* $\ell, u \in \mathbf{R}^A$ on the arcs and *costs* $c \in \mathbf{R}^A$ on the arcs.

We need some notation to talk about relaxed conservation. If $w \in \mathbf{R}^X$ and $Y \subseteq X$, then we abbreviate $\sum_{y \in Y} w_y$ by $w(Y)$ as usual. For node subset S, define $\Delta^+ S$ as $\{i \to j \in A \mid i \in S, j \notin S\}$, $\Delta^- S$ as $\{i \to j \in A \mid i \notin S, j \in S\}$, and $\Delta S = \Delta^+ S \cup \Delta^- S$. We say that arc *a crosses S* if $a \in \Delta S$. If φ is a flow on the arcs (i.e., $\varphi \in \mathbf{R}^A$), for $i \in N$ define the *boundary* $\partial\varphi(i)$ of φ at i to be $\varphi(\Delta^+\{i\}) - \varphi(\Delta^-\{i\})$, i.e., the net flow out of i. Thus $\partial\varphi$ is a vector in $\mathbf{R}^N$. We extend $\partial\varphi$ to node subsets $S \subseteq N$ as usual via $\partial\varphi(S) = \sum_{i \in S} \partial\varphi(i) = \varphi(\Delta^+ S) - \varphi(\Delta^- S)$, which equals the net flow out of S. Later we will consider several auxiliary networks whose arc sets are supersets of A, so we will often subscript ∂ and Δ by the set of arcs we want them to include at that point.

Usual MCF conservation requires that $\partial\varphi(i) = 0$ for all $i \in N$, which implies that $\partial\varphi(S) = 0$ for all $S \subseteq N$. From this perspective it is natural to relax this constraint to $\partial\varphi(S) \le f(S)$ for some set function $f : \mathcal{D} \to \mathbf{R}$ on some set family $\mathcal{D} \subseteq 2^N$ with $\emptyset, N \in \mathcal{D}$. Since $\partial\varphi(\emptyset) = \partial\varphi(N) = 0$ for any flow φ, we can and will assume that $f(\emptyset) = f(N) = 0$.

For such a set function f, the *base polyhedron* $\mathrm{B}(f)$ is defined by

$$\mathrm{B}(f) = \{x \mid x \in \mathbf{R}^N,\ x(N) = f(N) = 0,\ \forall S \in \mathcal{D} : x(S) \leq f(S)\}.$$

A vector in $\mathrm{B}(f)$ is called a *base*. This is defined so that $\partial\varphi(S) \leq f(S)$ for all $S \in \mathcal{D}$ is equivalent to requiring that $\partial\varphi \in \mathrm{B}(f)$. Then the (primal) optimization problem we would like to solve is the following:

$$\begin{aligned} &\text{Minimize} \sum_{a \in A} c_a \varphi_a \\ &\text{subject to } \ell_a \leq \varphi_a \leq u_a \qquad (a \in A), \\ &\partial\varphi \in \mathrm{B}(f). \end{aligned} \tag{2.1}$$

This is a linear program with an exponential number of constraints. Note that for general $\mathcal{D}$ and f, $\mathrm{B}(f)$ could be empty, and/or (2.1) could have fractional optimal solutions.

In order to ensure that $\mathrm{B}(f) \neq \emptyset$ and that we have integral optimal solutions, it is necessary to require some structure on $\mathcal{D}$ and f. For $S, T \subseteq N$, we say that S and T *intersect* if $S \cap T \neq \emptyset$, and *cross* if they intersect and also have $S \cup T \neq N$. We say that $\mathcal{D}$ is a *ring* (resp., *intersecting*, *crossing*) family if $S \cap T, S \cup T$ are also in $\mathcal{D}$ for all (resp., intersecting, crossing) pairs $S, T \in \mathcal{D}$ (note that a ring family is a distributive lattice). We say that f is a ring (resp., intersecting, crossing) *submodular* on $\mathcal{D}$ if $\mathcal{D}$ is a ring (resp., intersecting, crossing) family, and for all (resp., intersecting, crossing) pairs $S, T \in \mathcal{D}$ we have

$$f(S) + f(T) \geq f(S \cup T) + f(S \cap T). \tag{2.2}$$

Note that every ring family is also an intersecting and crossing family, and so every ring submodular f is also intersecting and crossing submodular.

Edmonds and Giles [4] originally formulated the SF problem for crossing submodular functions, which can have empty base polyhedra. Fujishige [14] provided a necessary and sufficient condition for $\mathrm{B}(f)$ to be nonempty. This condition can be checked efficiently by the bi-truncation algorithm of Frank and Tardos [12]. When f is crossing submodular on $\mathcal{D}$ and $\mathrm{B}(f) \neq \emptyset$, then (2.1) is an SF problem. Edmonds and Giles showed that submodular flow problems are totally dual integral (TDI), and so always have integral optimal solutions with integer data.

Furthermore, Fujishige [14] showed that a nonempty base polyhedron $\mathrm{B}(f)$ of a crossing submodular function f on a crossing family $\mathcal{D}$ is identical to a base polyhedron $\mathrm{B}(\widetilde{f})$ of a ring submodular function $\widetilde{f}$ on a ring family $\widetilde{\mathcal{D}}$ that contains $\mathcal{D}$, which was already implicit in Frank [9]. Therefore, theorems and algorithms concerning geometric properties of the base polyhedra of ring submodular functions carry over to crossing submodular functions.

The cut canceling algorithms presented in this paper rely only on geometric properties such as the exchange capacity (defined in section 3). This allows us to assume throughout most of the paper that $\mathcal{D}$ is a ring family and that f is ring submodular on $\mathcal{D}$. We sketch out the minor modifications needed to our algorithms to make them work for the crossing submodular case in section 8.

The base polyhedron of a ring submodular function is always nonempty. Technically, the dual problem to (2.1) should have an exponential number of dual variables, one for each subset of N. However, it is possible and much more convenient to simplify these to dual variables on nodes, or *node potentials* $\pi \in \mathbf{R}^N$. For arc $a = i \to j$, we define the *reduced cost* on a by $c_a^\pi = c_a + \pi_i - \pi_j$.

Node potentials are widely used in MCF algorithms, where they arise naturally as LP dual variables. The first use that we are aware of of node potentials as a simpler substitute for the dual variables for the set constraints in submodular problems is made by Iri and Tomizawa [23] for the independent assignment problem. The weight splitting in the weighted matroid intersection algorithm of Frank [8] plays essentially the same role as the node potentials. These two equivalent matroid optimization problems are special cases of the 0-1 SF problem, for which Frank [9] devised an efficient combinatorial algorithm using node potentials.

We need some further concepts to characterize optimality for (2.1). If φ is a flow with boundary $x = \partial\varphi$ such that $x \in \mathrm{B}(f)$, then we say that subset S is *φ-tight*, or *x-tight*, or just *tight* if $x(S)$ $(= \partial\varphi(S)) = f(S)$. Ring submodularity implies that the union and intersection of tight sets are also tight.

If π is a set of node potentials with distinct values $v_1 > v_2 > \cdots > v_h$, then the kth *level set* of π is $L_k^\pi \equiv \{i \in N \mid \pi_i \geq v_k\}$, $k = 1, 2, \ldots, h$. It will be convenient to let $L_0^\pi = \emptyset$. Note that $\emptyset = L_0^\pi \subset L_1^\pi \subset L_2^\pi \subset \cdots \subset L_h^\pi = N$. Suppose $L_k^\pi \in \mathcal{D}$ for $k = 0, 1, \ldots, h$, and define a function $f^\pi : \mathcal{D} \to \mathbf{R}$ by

$$f^\pi(S) = \sum_{k=1}^{h} \{f((S \cap L_k^\pi) \cup L_{k-1}^\pi) - f(L_{k-1}^\pi)\} \qquad (S \in \mathcal{D}).$$

Then f^π is ring submodular, $f^\pi \leq f$, and if S *nests* with each L_k^π (either $S \subseteq L_k^\pi$ or $L_k^\pi \subseteq S$), then $f^\pi(S) = f(S)$. Also, x is in $\mathrm{B}(f^\pi)$ if and only if $x \in \mathrm{B}(f)$ and every L_k^π is x-tight for f, which is true if and only if x maximizes $\pi^T y$ over $y \in \mathrm{B}(f)$ (see [16, Theorem 3.15]).

Cunningham and Frank [3] established an optimality criterion of the SF problem in terms of node potentials, which can be reformulated as follows. See also [16, Theorem 5.3] for a direct proof.

THEOREM 2.1. *An SF φ is optimal if and only if there exists a node potential $\pi : N \to \mathbf{R}$ such that*

$$c_a^\pi > 0 \Rightarrow \varphi_a = \ell_a,$$
$$c_a^\pi < 0 \Rightarrow \varphi_a = u_a,$$

and $\partial\varphi \in \mathrm{B}(f^\pi)$.

Thus two equivalent ways to state the "submodular part" of the optimality condition are to require that an optimal φ and π satisfy that each L_k^π is φ-tight, or that $\partial\varphi$ maximizes the objective $\pi^T y$ over $y \in \mathrm{B}(f)$.

3. Dual approximate optimality. We first cover the details of how to check node potentials π for optimality. We then show how to adapt checking exact optimality to checking a new kind of approximate optimality. This checking routine will form the core of our cut canceling algorithm, as it will produce the cuts that we cancel. For this paper a *cut* is just a nonempty, proper subset $S \in \mathcal{D}$ of N. We cancel a cut by increasing π_i by some step length β for $i \in S$, which has the effect of increasing c_a^π on $\Delta^+ S$, decreasing c_a^π on $\Delta^- S$, and changing the level sets of π.

Given a node potential π, we define *modified bounds* ℓ^π and u^π as follows. If $c_a^\pi < 0$, then $\ell_a^\pi = u_a^\pi = \ell_a$; if $c_a^\pi = 0$, then $\ell_a^\pi = \ell_a$ and $u_a^\pi = u_a$; if $c_a^\pi > 0$, then $\ell_a^\pi = u_a^\pi = u_a$. Then Theorem 2.1 implies that π is optimal if and only if there is a feasible flow in the network G^π with bounds ℓ^π, u^π, and f^π.

For the proof of correctness of our algorithm we will need some details of how feasibility of G^π is checked. We do this using a variant of an algorithm of Frank [10] that we call the *feasibility algorithm.* It starts with any initial base $x \in \mathrm{B}(f^\pi)$ and any initial φ satisfying ℓ^π and u^π. We now define a *residual network* on N. If $\varphi_{ij} > \ell^\pi_{ij}$, then we have a backward residual arc $j \to i$ with *residual capacity* $r_{ij} = \varphi_{ij} - \ell^\pi_{ij}$; if $\varphi_{ij} < u^\pi_{ij}$, then we have a forward residual arc $i \to j$ with $r_{ji} = u^\pi_{ij} - \varphi_{ij}$. To deal with relaxed conservation, for every $i, j \in N$ with $i \neq j$ and $x \in \mathrm{B}(f)$, we define the *exchange capacity* w.r.t. x as

$$\widetilde{r}(x, i, j) = \max\{\alpha \mid x + \alpha\chi_i - \alpha\chi_j \in \mathrm{B}(f)\};$$

thus $\widetilde{r}(x, i, j) > 0$ means that there is no x-tight set containing i but not j. If we compute exchange capacity relative to f^π instead of f, then we denote it by $\widetilde{r}^\pi$. Since $f^\pi \le f$, we have $\widetilde{r}^\pi(x, i, j) \le \widetilde{r}(x, i, j)$. Make a *jumping arc* $j \to i$ with capacity $\widetilde{r}(x, i, j)$ whenever $\widetilde{r}(x, i, j) > 0$. Note that S is x-tight if and only if there are no jumping arcs w.r.t. x entering S. The residual network for π contains only the jumping arcs w.r.t. $\widetilde{r}^\pi$. Also, [3, Theorem 7] shows that for any node potentials $\pi \in \mathbf{R}^N$ and $i, j \in N$ with $\pi_i = \pi_j$, the exchange capacity $\widetilde{r}^\pi(x, i, j)$ for f^π is determined by a set S that nests with the L^π_k so that $\widetilde{r}^\pi(x, i, j) = f^\pi(S) - x(S) = f(S) - x(S) \ge \widetilde{r}(x, i, j)$, and hence $\widetilde{r}^\pi(x, i, j) = \widetilde{r}(x, i, j)$.

The feasibility algorithm finds directed residual paths from $N^+ = \{i \in N \mid x_i > \partial\varphi(i)\}$ to $N^- = \{i \in N \mid x_i < \partial\varphi(i)\}$ with a minimum number of arcs in the residual network. On each path it augments flow φ on the residual arcs, and modifies x as per the jumping arcs, which monotonically reduces the difference of x and $\partial\varphi$. By using a lexicographic selection rule, the algorithm terminates in finite time. At termination, either x coincides with $\partial\varphi$, which implies the optimality of π, or there is no directed path from N^+ to N^-.

In this last case, define $T \subseteq N$ as the set of nodes from which N^- is reachable by directed residual paths. No jumping arcs enter T, so it must belong to $\mathcal{D}$, it must be tight for the final x, and it must contain all i with $x_i < \partial\varphi(i)$. Furthermore, we have $\varphi(\Delta^+T) = \ell^\pi(\Delta^+T)$ and $\varphi(\Delta^-T) = u^\pi(\Delta^-T)$. Thus we get

$$V^\pi(T) \equiv \ell^\pi(\Delta^+_A T) - u^\pi(\Delta^-_A T) - f^\pi(T) = \partial\varphi(T) - x(T) > 0. \tag{3.1}$$

We call a node subset S with $V^\pi(S) > 0$ a *positive cut.* Similar reasoning shows that for any other $S \in \mathcal{D}$, we have $V^\pi(S) \le \partial\varphi(S) - x(S) \le \partial\varphi(T) - x(T) = V^\pi(T)$, proving that T is an *MPC.* Intuitively, $V^\pi(T)$ measures how far away from optimality π is. We summarize as follows.

Lemma 3.1. *Node potentials π are optimal if and only if there are no positive cuts w.r.t. π. When π is not optimal, the output of the feasibility algorithm is an MPC.*

We denote the running time of the feasibility algorithm by FA. As usual, we assume that we have an oracle to compute exchange capacities and denote its running time by h. Computing an exchange capacity is an SFM on a distributive lattice, which can be done via the ellipsoid method [21] or by combinatorial methods [25, 41] in strongly polynomial time. Cunningham and Frank [3, Theorem 7] show how to derive an oracle for computing exchange capacities for f^π from an oracle for f with the same running time. Fujishige and Zhang [17] (see also [27]) show how to extend the push-relabel algorithm of Goldberg and Tarjan [19] to get $\mathrm{FA} = \mathrm{O}(n^3 h)$.

Unfortunately, even for min-cost flow, an example of Hassin [22] shows that it may be necessary to cancel an exponential number of MPCs to achieve optimality. In [42] we get around this difficulty for MCF by relaxing the optimality conditions by a parameter $\delta > 0$, and applying scaling to δ. It then turns out that only a polynomial number of relaxed MPCs need to be canceled for a given value of the parameter, and only a polynomial number of scaled subproblems need to be solved.

Until now it has been difficult to find a relaxation in the SF case that would allow the same analysis to apply. Here we introduce a new relaxation that works. We relax the modified bounds on the arcs in A by widening them by δ just as in [5]. Define $\ell_a^{\pi,\delta} = \ell_a^{\pi} - \delta$ and $u_a^{\pi,\delta} = u_a^{\pi} + \delta$ for every arc $a \in A$.

We also need to relax the submodular bounds on conservation by some function of δ. Define $f^{\pi,\delta}(S) = f^{\pi}(S) + \delta|S| \cdot |N - S|$, which is closely related to a relaxation introduced in [24]. Since $|S| \cdot |N - S|$ is submodular, $f^{\pi,\delta}$ is submodular.

We can now define the relaxed cut value of S as

$$V^{\pi,\delta}(S) = \ell^{\pi,\delta}(\Delta_A^+ S) - u^{\pi,\delta}(\Delta_A^- S) - f^{\pi,\delta}(S).$$

We say that node potential π is *δ-optimal* if $V^{\pi,\delta}(S) \leq 0$ holds for every cut S. Thus π is 0-optimal if and only if it is optimal.

To check if π is δ-optimal, we need to see if there is a feasible flow in the network with bounds $\ell^{\pi,\delta}$, $u^{\pi,\delta}$, and $f^{\pi,\delta}$. It turns out to be more convenient to move the δ relaxation off $f^{\pi,\delta}$ onto a new set of arcs. Define $\widehat{G}^{\pi} = (N, A \cup E)$ to be the directed graph obtained from G by adding $E = \{i \to j \mid i, j \in N,\ i \neq j\}$ to the arc set. Extend the bounds ℓ^{π}, u^{π}, $\ell^{\pi,\delta}$, and $u^{\pi,\delta}$ from A to E by setting $\ell_e^{\pi} = u_e^{\pi} = 0$, $\ell_e^{\pi,\delta} = 0$, and $u_e^{\pi,\delta} = \delta$ for every $e \in E$. For convenience set $I = A \cup E$ and $m' = |A \cup E| = m + n(n-1)$.

Now checking $\ell^{\pi,\delta}$, $u^{\pi,\delta}$, and $f^{\pi,\delta}$ for feasibility on G (with arc set A) is equivalent to checking $\ell^{\pi,\delta}$, $u^{\pi,\delta}$, and f^{π} for feasibility on $\widehat{G}^{\pi}$ (with arc set $I = A \cup E$). Also, the relaxed cut value on G can be re-expressed as

$$V^{\pi,\delta}(S) = \ell^{\pi,\delta}(\Delta_I^+ S) - u^{\pi,\delta}(\Delta_I^- S) - f^{\pi}(S).$$

If there is a feasible SF in $\widehat{G}^{\pi}$, then π is δ-optimal. Otherwise, the feasibility algorithm gives us a node subset T that maximizes $V^{\pi,\delta}(S)$, a *relaxed MPC*, or *δ-MPC*. When π is not δ-optimal, the feasibility algorithm also gives us an optimal "max flow" φ on $\widehat{G}^{\pi}$ and a base x in $\mathrm{B}(f^{\pi})$ such that $x_i \leq \partial_I \varphi(i)$ for $i \in T$ and $x_i \geq \partial_I \varphi(i)$ for $i \notin T$.

We also need to consider max mean cuts. The *mean value* of cut S is

$$\overline{V}^{\pi}(S) \equiv \frac{V^{\pi}(S)}{|\Delta_A S| + |S| \cdot |N - S|}.$$

A max mean cut attains the maximum in $\delta(\pi) \equiv \max_S \overline{V}^{\pi}(S)$. By standard LP duality arguments, $\delta(\pi)$ also equals the minimum δ such that there is a feasible flow in $\widehat{G}^{\pi}$ with bounds $\ell^{\pi,\delta}$, $u^{\pi,\delta}$, and f^{π}. Define U to be the maximum absolute value of any ℓ_a, u_a, or $f(\{i\})$. A max mean cut can be computed using $\mathrm{O}(\min\{m', \frac{\log(nU)}{1+\log\log(nU)-\log\log n}\})$ calls to the feasibility algorithm in the framework of discrete Newton's algorithm; see McCormick and Ervolina [34] or Radzik [37, 38].

4. Cut cancellation.

We will start out with δ large and drive δ towards zero, since π is 0-optimal if and only if it is optimal. The next lemma says that δ need not start out too big, and need not end up too small. Its proof is similar to [5, Lemma 5.1].

LEMMA 4.1. *Suppose that ℓ, u, and f are integral. Then any node potentials π are $2U$-optimal. Moreover, when $\delta < 1/m'$, any δ-optimal node potentials are optimal.*

Our relaxed δ-MPC canceling algorithm will start with $\delta = 2U$ and will execute scaling phases, where each phase first sets $\delta := \delta/2$. The input to a phase will be a

2δ-optimal set of node potentials from the previous phase, and its output will be a δ-optimal set of node potentials. Lemma 4.1 says that after $\mathrm{O}(\log(nU))$ scaling phases we have $\delta < 1/m'$ and we are optimal. Within a scaling phase we use the feasibility algorithm to find a δ-MPC T. We then *cancel* T by adding a constant *step length* β to π_i for each node in T to get π'.

In ordinary min-cost flow we choose the step length β based on the first arc in A, whose reduced cost hits zero (as long as the associated flow is within the bounds; see [42, Figure 2]). This bound on β is

$$\eta \equiv \min \left\{ \begin{array}{l} \min\{|c_a^\pi| \mid a \in \Delta_A^+ T,\, c_a^\pi < 0,\, \varphi_a \geq \ell_a\} \\ \min\{c_a^\pi \mid a \in \Delta_A^- T,\, c_a^\pi > 0,\, \varphi_a \leq u_a\} \end{array} \right\}.$$

(Note that optimality of T implies that every arc a with $\varphi_a \geq \ell_a$ has negative reduced cost, and every arc a with $\varphi_a \leq u_a$ has positive reduced cost.)

Here we must also worry about the level set structure of π changing during the cancel, which is equivalent to the reduced cost of a jumping arc reaching zero. We need to increase flows on jumping arcs leaving T so that the $L_k^{\pi'}$ will be tight. We try to do this by decreasing flow on arcs of $\Delta_E^- T$, all of which have flow δ since T is tight. If the exchange capacity of such an arc is at most δ we can do this. Otherwise, this E-arc will determine β, and the large exchange capacity will allow us to show that the algorithm makes sufficient progress.

These considerations lead to the subroutine AdjustFlow below. It takes as input the optimal max flow φ from the feasibility algorithm, and its associated base $x \in \mathrm{B}(f^\pi)$. It computes an update $\psi \in \mathbf{R}^E$ to φ and base x' so that x' will be the base associated with $\varphi' \equiv \varphi - \psi$. Note that in AdjustFlow we always compute $\widetilde{r}(x, i, j)$ w.r.t. f, never w.r.t. f^π, so that jumping arcs are w.r.t. f, not f^π. We call an arc that determines β a *determining arc.*

```
Algorithm AdjustFlow:
    begin
        $\psi_e = 0$ for $e \in E$;
        $H := \{j \to i \mid i \in T,\, j \in N - T,\, 0 < \pi_j - \pi_i < \eta\}$;
        while $H \neq \emptyset$ do
        begin
            $x' := x - \partial_E \psi$;
            select $e = j \to i \in H$ with minimum $\pi_j - \pi_i$;
            set $H := H - \{e\}$;
            if $\widetilde{r}(x', i, j) < \delta$ then $\psi_e := \widetilde{r}(x', i, j)$;
            else return $\beta := \pi_j - \pi_i$    [$\beta$ is determined by jumping arc $j \to i$];
        end
        return $\beta := \eta$    [$\beta$ is determined by the $A$-arc determining $\eta$];
    end.
```

The full description of canceling a cut is now: Use the feasibility algorithm to find δ-MPC T, max flow φ, and base x. Run AdjustFlow to modify φ to $\varphi' = \varphi - \psi$ and x to $x' = x - \partial_E \psi$, and to choose β. Finally, increase π_i by β for all $i \in T$. A scaling phase cancels δ-MPCs in this manner until π is δ-optimal. The next results show that canceling a δ-MPC cannot increase the δ-MPC value, and that $V^{\pi,\delta}(S)$ will decrease by at least δ under some circumstances.

Lemma 4.2. *Suppose we cancel a δ-MPC T for π to get a node potential π'. Then $V^{\pi',\delta}(S) \leq V^{\pi,\delta}(T)$ holds for any cut S.*

Proof. It is easy to show, similar to [42, Lemma 5.3], that $\ell_a^{\pi',\delta} \le \varphi_a \le u_a^{\pi',\delta}$ holds for every $a \in A$. As a result of AdjustFlow, we obtain a flow $\psi \in \mathbf{R}^E$ with $0 \le \psi_e < \delta$ for $e \in E$. Put $\varphi'_a = \varphi_a$ for $a \in A$ and $\varphi'_e = \varphi_e - \psi_e$ for $e \in E$. To prove that φ' is feasible for $\mathrm{B}(f^{\pi'})$ we need the following.

Claim: $x' = x - \partial_E \psi \in \mathrm{B}(f^{\pi'})$.

Proof. If the initial $H = \emptyset$ in AdjustFlow (i.e., $\pi_j > \pi_i$ implies that $\pi_j \ge \pi_i + \eta$), then $\psi \equiv 0$ and so $x' = x$. In this case it can be shown that $f^{\pi'}(S) = f^{\pi}(S \cap T) + f^{\pi}(S \cup T) - f^{\pi}(T)$. Since $x(T) = x'(T) = f^{\pi}(T)$, we have $x'(S) = x'(S \cap T) + x'(S \cup T) - x'(T) \le f^{\pi}(S \cap T) + f^{\pi}(S \cup T) - f^{\pi}(T) = f^{\pi'}(S)$.

Suppose instead that the initial H is nonempty. Here we know at least that $x' \in \mathrm{B}(f)$ (since ψ_{ji} is never bigger than $\widetilde{r}(x', i, j)$). Denote $T \cap (L_k^{\pi} - L_{k-1}^{\pi})$ by T_k and $(L_k^{\pi} - L_{k-1}^{\pi}) - T$ by $\overline{T}_k$, so that the T_k partition T and the $\overline{T}_k$ partition $N - T$. Then a typical level set of π' looks like $L' = (\bigcup_{k=1}^{q} T_k) \cup (\bigcup_{k=1}^{p} \overline{T}_k) = L_p^{\pi} \cup (T \cap L_q^{\pi})$ for some $0 \le p < q \le h$.

Now both T and L_i^{π} are φ-tight for f^{π}, so their union and intersection are both φ-tight for f^{π}. By the same reasoning, $\widehat{T}_i \equiv (T \cap L_i^{\pi}) \cup L_{i-1}^{\pi} = (\bigcup_{k=1}^{i} T_k) \cup (\bigcup_{k=1}^{i-1} \overline{T}_k)$ is also φ-tight for f^{π}. Since every arc of H starts in some $\overline{T}_j$ and ends in some T_i with $j < i$, no arc of H crosses any $\widehat{T}_i$; thus each $\widehat{T}_i$ is also φ'-tight for f^{π}. Each $\widehat{T}_i$ nests with the L_k^{π}, so we have $f^{\pi}(\widehat{T}_i) = f(\widehat{T}_i)$, and thus each $\widehat{T}_i$ is in fact φ'-tight for f. This implies that the only possible jumping arcs entering level set L' of π' are those from $\overline{T}_j$ to T_i for $p < j < i \le q$. But these jumping arcs all belong to H and were removed by AdjustFlow. Thus L' is φ'-tight for f, and so $x' \in \mathrm{B}(f^{\pi'})$. □

Since $\psi_e > 0$ implies $\varphi_e = \delta$, every $e \in E$ satisfies $0 \le \varphi'_e \le \delta$. Therefore we have

$$\begin{aligned}
V^{\pi',\delta}(S) &= \ell^{\pi',\delta}(\Delta_I^+ S) - u^{\pi',\delta}(\Delta_I^- S) - f^{\pi'}(S) && \text{(definition of } V^{\pi',\delta}(S)\text{)}\\
&\le \partial_I \varphi'(S) - x'(S) && \text{(feasibility of } \varphi',\ x' \in \mathrm{B}(f^{\pi'})\text{)}\\
&= \partial_I \varphi(S) - x(S) && \text{(definition of } \varphi',\ x'\text{)}\\
&\le \partial_I \varphi(T) - x(T) && (T \text{ a } \delta\text{-MPC})\\
&= \ell^{\pi,\delta}(\Delta_I^+ T) - u^{\pi,\delta}(\Delta_I^- T) - f^{\pi}(T) && (T \text{ is } \varphi\text{-tight for } f^{\pi})\\
&= V^{\pi,\delta}(T) && \text{(definition of } V^{\pi,\delta}(T)\text{)}. \qquad □
\end{aligned}$$

Corollary 4.3. *With the same hypothesis and notation as Lemma* 4.2*, if the determining arc $a \in A \cup E$ crosses S, then we have $V^{\pi',\delta}(S) \le V^{\pi,\delta}(T) - \delta$.*

Proof. If the determining arc $a \in A$, then $\ell_a^{\pi',\delta} + \delta = \ell_a \le \varphi_a \le u_a = u_a^{\pi',\delta} - \delta$, and hence $\ell^{\pi',\delta}(\Delta_I^+ S) - u^{\pi',\delta}(\Delta_I^- S) \le \partial_I \varphi'(S) - \delta$. If $a \in \Delta_E^+ S$, then $\varphi_a = \delta = \ell_a^{\pi',\delta} + \delta$, and hence $\ell^{\pi',\delta}(\Delta_I^+ S) - u^{\pi',\delta}(\Delta_I^- S) \le \partial_I \varphi'(S) - \delta$. If $a = j \to i \in \Delta_E^- S$, then $f^{\pi'}(S) - x'(S) \ge \widetilde{r}^{\pi'}(x', i, j)$. Now a determining means that $\pi'_i = \pi'_j$, which implies $\widetilde{r}^{\pi'}(x', i, j) = \widetilde{r}(x', i, j) \ge \delta$, and hence $f^{\pi'}(S) \ge x'(S) + \delta$. In all cases we have $V^{\pi',\delta}(S) \le \partial_I \varphi'(S) - x'(S) - \delta$, which implies $V^{\pi',\delta}(S) \le V^{\pi,\delta}(T) - \delta$. □

Note that we have two different notions of "closeness to optimality" in this algorithm. At the outer level of scaling phases we drive δ towards zero (and so π towards optimality), and inside a phase we drive the δ-MPC value towards zero (and so π towards δ-optimality). Observe that the difference between $\partial\varphi$ and x in the feasibility algorithm is an inner relaxation of the condition that π is δ-optimal, in that we keep a flow satisfying the bounds $\ell^{\pi,\delta}$ and $u^{\pi,\delta}$ and a base x in $\mathrm{B}(f^{\pi})$. As the phase progresses the difference is driven to zero, until the final flow proves δ-optimality of the final π.

We can now prove the key lemma in the convergence proof of δ-MPC canceling, which shows that we must be in a position to apply Corollary 4.3 reasonably often.

Lemma 4.4. *After at most n cut cancellations, the value of a δ-MPC decreases by at least δ.*

Proof. Suppose that the first cancellation has initial node potentials π^0 and cancels δ-MPC T^1 to get π^1, the next cancels T^2 to get π^2, …, and the nth cancels T^n to get π^n. Each cancellation makes at least one arc into a determining arc. Consider the subgraph of these determining arcs. If a cut creates a determining arc whose ends are in the same connected component of this subgraph, then this cut must be crossed by a determining arc from an earlier cut. We can avoid this only if each new determining arc strictly decreases the number of connected components in the subgraph. This can happen at most $n - 1$ times, so it must happen at least once within n iterations.

Let k be an iteration where T^k shares a determining arc with an earlier cut T^h. By Corollary 4.3 applied to $T = T^h$ and $S = T^k$, we have $V^{\pi^{h-1},\delta}(T^h) \geq V^{\pi^h,\delta}(T^k) + \delta$. Note that the δ-MPC value at iteration i is $V^{\pi^{i-1},\delta}(T^i)$. If $V^{\pi^h,\delta}(T^k) \geq V^{\pi^{k-1},\delta}(T^k)$, Lemma 4.2 says that $V^{\pi^0,\delta}(T^1) \geq V^{\pi^{h-1},\delta}(T^h) \geq V^{\pi^h,\delta}(T^k) + \delta \geq V^{\pi^{k-1},\delta}(T^k) + \delta \geq V^{\pi^{n-1},\delta}(T^n) + \delta$.

If instead $V^{\pi^h,\delta}(T^k) < V^{\pi^{k-1},\delta}(T^k)$, then let p be the latest iteration between h and k with $V^{\pi^{p-1},\delta}(T^k) < V^{\pi^p,\delta}(T^k)$. The only way for the value of T^k to increase like this is if there is an arc $a \in A$ with $c_a^{\pi^{p-1}} = 0$ that crosses T^k in the reverse orientation to its orientation in T^p, or $f^{\pi^{p-1},\delta}(T^k) > f^{\pi^p,\delta}(T^k)$ holds. Let φ^p be a flow proving optimality of T^p. If $a \in \Delta_A^+ T^p \cap \Delta_A^- T^k$ ($a \in \Delta_A^- T^p \cap \Delta_A^+ T^k$), then we have $u_a^{\pi^p,\delta} = \varphi_a^p + 2\delta \geq \varphi_a^p + \delta$ ($l_a^{\pi^p,\delta} = \varphi_a^p - 2\delta \leq \varphi_a^p - \delta$). When $f^{\pi^{p-1},\delta}(T^k) > f^{\pi^p,\delta}(T^k)$ holds, there exist $i, j \in N$ such that $i \in T^k \setminus T^p$, $j \in T^p \setminus T^k$. It follows from $i \to j \in \Delta_I^- T^p$ and $j \to i \in \Delta_I^+ T^p$ that $\varphi_e^p = \delta = l_e^{\pi^p,\delta} + \delta$ for $e = i \to j$ and $\varphi_{e'}^p = 0 = u_{e'}^{\pi^p,\delta} - \delta$ for $e' = j \to i$. In either case, we have $l^{\pi^p,\delta}(\Delta_I^+ T^k) - u^{\pi^p,\delta}(\Delta_I^- T^k) \leq \partial_I \varphi^p(T^k) - \delta$. Thus equations in the proof of Lemma 4.2 apply, showing that $V^{\pi^{p-1},\delta}(T^p) - \delta \geq V^{\pi^p,\delta}(T^k)$. Then Lemma 4.2 and the choice of p say that $V^{\pi^0,\delta}(T^1) \geq V^{\pi^{p-1},\delta}(T^p) \geq V^{\pi^p,\delta}(T^k) + \delta \geq V^{\pi^{k-1},\delta}(T^k) + \delta \geq V^{\pi^{n-1},\delta}(T^n) + \delta$. □

Lemma 4.5. *The value of every δ-MPC in a scaling phase is at most $m'\delta$.*

Proof. Let φ prove the 2δ-optimality of the initial π in a phase, so that φ violates the bounds l^π and u^π by at most 2δ on every arc of $A \cup E$, and $\varphi_{si} = 0$ for all i. To get an initial flow φ^0 to begin the next phase that violates the bounds by at most δ, we need to change φ by at most δ on each arc of $A \cup E$. Then if we set $\varphi_{si}^0 = \delta|\Delta_I^+\{i\}|$ for all i, φ^0 will satisfy f^π also. The sum $\sum_{i \in N} \varphi_{si}^0$ is at most $m'\delta$, and this is an upper bound on the value of the first δ-MPC in this phase. By Lemma 4.2, it is then also a bound on the value of every δ-MPC in the phase. □

Putting Lemmas 4.4 and 4.5 together yields our first bound on the running time of δ-MPC canceling.

Theorem 4.6. *A scaling phase of δ-MPC canceling cancels at most $m'n$ δ-MPCs. Thus the running time of δ-MPC canceling is*

$$\mathrm{O}(n^3 \log(nU)\mathrm{FA}) = \mathrm{O}(n^6 h \log(nU)).$$

Proof. Lemma 4.5 shows that the δ-MPC value of the first cut in a phase is at most $m'\delta$. It takes at most n iterations to reduce this by δ, so there are at most $m'n = \mathrm{O}(n^3)$ iterations per phase. The time per iteration is dominated by computing a δ-MPC, which is O(FA). The number of phases is $\mathrm{O}(\log(nU))$ by Lemma 4.1. □

5. A strongly polynomial bound. We first prove the following lemma. Such a result was first proved by Tardos [43] for MCF; this version is an extension of dual versions by Fujishige [15] and Ervolina and McCormick [5].

LEMMA 5.1. *Suppose that flow φ' proves that π' is δ'-optimal. If arc $a \in A$ satisfies $\varphi'_a < u_a - (m'+1)\delta'$, then all optimal π^* have $c_a^{\pi^*} \geq 0$. If arc $a \in A$ satisfies $\varphi'_a > \ell_a + (m'+1)\delta'$, then all optimal π^* have $c_a^{\pi^*} \leq 0$. If $\partial_I \varphi'(T^0) > f^{\pi^0}(T^0) + m'n\delta'$ for some π^0 and some $T^0 \in \mathcal{D}$, then there is an E-arc $i \to j$ leaving some level set of π^0 such that all optimal π^* have $\pi_i^* \leq \pi_j^*$.*

Proof. Note that φ' proving δ'-optimality of π' implies that there is no jumping arc entering any $L_k^{\pi'}$ for any k. Since $x' \equiv \partial_I \varphi' \in \mathrm{B}(f^{\pi'})$, this says that x' is a π'-maximum base of $\mathrm{B}(f)$.

Now change φ', a flow on $A \cup E$ feasible for $\ell^{\pi',\delta'}$ and $u^{\pi',\delta'}$, into flow $\widehat{\varphi}$ on just A, feasible for $\ell^{\pi'}$ and $u^{\pi'}$, by getting rid of φ'_e for $e \in E$ and by changing φ'_a by at most δ' on each $a \in A$. Note that this $\widehat{\varphi}$ will not, in general, satisfy $\partial_A \widehat{\varphi} \in \mathrm{B}(f)$, but that $\widehat{\varphi}$ otherwise satisfies all optimality conditions: it satisfies the bounds and is complementary slack with π'. Define $N^+ = \{i \in N \mid x'_i > \partial_A \widehat{\varphi}(i)\}$ and $N^- = \{i \in N \mid x'_i < \partial_A \widehat{\varphi}(i)\}$ so that

$$x'(N^+) \leq m'\delta' + \partial_A \widehat{\varphi}(N^+). \tag{5.1}$$

We now apply the successive shortest path algorithm for submodular flow of Fujishige [13] starting from $\widehat{\varphi}$. (As originally stated, this algorithm is finite only for integer data, but the lexicographic techniques of Schönsleben [39] and Lawler and Martel [31] show that it can be made finite for any data.) This algorithm looks for an augmenting path from a node $i \in N^+$ to a node $j \in N^-$, where residual capacities on A-arcs come from $\widehat{\varphi}$ and residual capacities on jumping arcs come from x'. It chooses a shortest augmenting path (using the current reduced costs as lengths; such a path can be shown to always exist) and augments flow along the path, updating π' by the shortest path distances and x' as per the jumping arcs. This update maintains the properties that the current $\widehat{\varphi}$ satisfies the bounds and is complementary slack with the current π', and the current x' belongs to $\mathrm{B}(f)$. The algorithm terminates with optimal flow φ^* once the boundary of the current $\widehat{\varphi}$ coincides with the current x'. By (5.1), the total amount of flow pushed by this algorithm is at most $m'\delta'$.

This implies that for each $a \in A$, $\widehat{\varphi}_a$ differs from φ^*_a by at most $m'\delta'$, so φ'_a differs from φ^*_a by at most $(m'+1)\delta'$. In particular, if $\varphi'_a < u_a - (m'+1)\delta'$, then $\varphi^*_a < u_a$, implying that $c_a^{\pi^*} \geq 0$, and similarly for $\varphi'_a > \ell_a + (m'+1)\delta'$.

Suppose that P is an augmenting path chosen by the algorithm and that flow is augmented by amount τ_P along P. Since P has at most $n-1$ jumping arcs, the boundary of any $S \subseteq N$ changes by at most $(n-1)\tau_P$ due to P. Since $\sum_P \tau_P \leq m'\delta'$ by (5.1), the total change in $\partial_A \widehat{\varphi}(S)$ during the algorithm is at most $m'(n-1)\delta'$. Since $|\partial_I \varphi'(S) - \partial_A \widehat{\varphi}(S)| \leq m'\delta'$, the total change in $\partial_I \varphi'(S)$ is at most $m'n\delta'$. Thus $\partial_I \varphi'(T^0) > f^{\pi^0}(T^0) + m'n\delta'$ implies that $\partial_A \varphi^*(T^0) > f^{\pi^0}(T^0)$. This says that some level set of π^0 is not φ^*-tight. This implies that there is some E-arc $i \to j$ with $\pi^0_i > \pi^0_j$ but $\pi^*_i \leq \pi^*_j$. □

We now modify our algorithm a little bit. We divide our scaling phases into *blocks* of $\log_2(m'+1)$ phases each. At the beginning of each block we compute a max mean cut T^0 with mean value $\delta^0 = \delta(\pi^0)$ and cancel T^0 (including calling ADJUSTFLOW). This ensures that our current flow is δ^0-optimal, so we set $\delta = \delta^0$ and start the block of scaling phases. It will turn out that only $2m'$ blocks are sufficient to attain optimality.

Theorem 5.2. *This modified version of the algorithm takes* $\mathrm{O}(n^5 \log n \mathrm{FA}) = \mathrm{O}(n^8 h \log n)$ *time.*

Proof. Let T^0 be the max mean cut canceled at the beginning of the block, with associated node potential π^0, flow φ^0, and mean value δ^0. Complementary slackness for T^0 implies that $\varphi^0_a = u^{\pi^0}_a + \delta^0$ for all $a \in \Delta^-_I T^0$, that $\varphi^0_a = \ell^{\pi^0}_a - \delta^0$ for all $a \in \Delta^+_A T^0$, that $\varphi^0_a = \ell^{\pi^0}_a$ for all $a \in \Delta^+_I T^0$, and that $\partial_I \varphi^0(T^0) = f^{\pi^0}(T^0)$. Let π', φ', and δ' be the similar values after the last phase in the block. Since each scaling phase cuts δ in half, we have $\delta' < \delta^0/(m'+1)$.

Subtracting $\varphi'(\Delta^+_I T^0) - \varphi'(\Delta^-_I T^0) - \partial_I \varphi'(T^0) = 0$ from $V^{\pi^0}(T^0)$ yields

$$\begin{aligned}(|\Delta_A T^0| + |T^0| \cdot |N - T^0|)\delta^0 &= V^{\pi^0}(T^0)\\ &= (\ell^{\pi^0} - \varphi')(\Delta^+_I T^0) + (\varphi' - u^{\pi^0})(\Delta^-_I T^0)\\ &\quad + (\partial_I \varphi'(T^0) - f^{\pi^0}(T^0)).\end{aligned} \tag{5.2}$$

Now apply Lemma 5.1 to φ' and π'. If the term for arc a of $(\ell^{\pi^0} - \varphi')(\Delta^+_A T^0)$ is at least $\delta^0 > (m'+1)\delta'$, then we must have that $\ell^{\pi^0}_a = u_a$. Therefore $\varphi'_a < u_a - (m'+1)\delta'$, and we can conclude that $c^{\pi^*}_a \geq 0$. But each a in $\Delta^+_A T^0$ had $c^{\pi^0}_a < 0$, so this is a new sign constraint on c^π. The case for terms of $(\varphi' - u^{\pi^0})(\Delta^-_A T^0)$ is similar.

Suppose instead that all the terms in the $\Delta^+_A T^0$ and $\Delta^-_A T^0$ sums of (5.2) are at most δ^0. The total of all the E-arc terms is at most $|T^0| \cdot |N - T^0|\delta'$. Therefore the only possibility left to achieve the large left-hand side of (5.2) is to have $\partial_I \varphi'(T^0) > f^{\pi^0}(T^0) + (m'n/2)\delta'$. Lemma 5.1 says that in this case there must be a jumping arc $i \to j$ leaving some level set of π^0 such that all optimal π^* have $\pi^*_i \leq \pi^*_j$. Since π^0_i was larger than π^0_j, this is a new sign restriction on π.

In either case each block imposes a new sign restriction on $c^{\pi^*}_a$ for some I-arc a. At most $2m'$ such sign restrictions can be imposed before π^* is completely determined, so after at most $2m' = \mathrm{O}(n^2)$ blocks we must be optimal. Each block requires $\log(m'+1) = \mathrm{O}(\log n)$ scaling phases. The proof of Theorem 4.6 shows that each scaling phase costs $\mathrm{O}(n^3\mathrm{FA})$ time exclusive of computing the max mean cut. The time for computing a max mean cut is $\mathrm{O}(n^2\mathrm{FA})$, which is not a bottleneck. □

6. Separable convex cost submodular flow.

6. Separable convex cost submodular flow. This section is devoted to a straightforward extension of our algorithm to the separable convex cost SF problem. Of course, the linear case is a special case of the separable convex case, but this section is easier to understand after seeing the detailed proofs for the linear case of the previous section.

Let $g_a : \mathbf{R} \to \mathbf{R}$ be a convex cost function for an arc $a \in A$. Then the convex cost SF problem is as follows:

$$\begin{aligned}&\text{Minimize} \sum_{a \in A} g_a(\varphi_a)\\ &\text{subject to } \partial\varphi \in \mathrm{B}(f).\end{aligned}$$

(Here we take advantage of the ability to "hide" bounds in convex objective functions.) We denote by $g^{\dashv}_a(\xi)$ and $g^{\vdash}_a(\xi)$, respectively, the left derivative and the right derivative of g_a at $\xi \in \mathbf{R}$. Then an optimality condition for this problem is as follows.

Theorem 6.1 (see [16, Theorem 12.1]). *An SF* φ *is optimal if and only if there exists node potentials* $\pi : N \to \mathbf{R}$ *such that* $g^{\dashv}_a(\varphi_a) \leq \pi_j - \pi_i \leq g^{\vdash}_a(\varphi_a)$ *for every arc* $a = i \to j$ *and* $\partial\varphi \in \mathrm{B}(f^\pi)$. □

We define the modified bounds of $a = i \rightarrow j$ by

$$u_a^\pi = \sup\{x \mid g_a^\dashv(x) \leq \pi_j - \pi_i\}, \quad \ell_a^\pi = \inf\{x \mid g_a^\vdash(x) \geq \pi_j - \pi_i\}.$$

Then Theorem 6.1 implies that π is optimal if and only if there is a feasible flow in the network G^π with bounds ℓ^π, u^π, and f^π. From ℓ^π and u^π we get $\ell^{\pi,\delta} = \ell^\pi - \delta$ and $u^{\pi,\delta} = u^\pi + \delta$ as before. Using $\ell^{\pi,\delta}$ and $u^{\pi,\delta}$ we define δ-MPCs and perform the δ-MPC canceling algorithm just as in the linear cost case. The preliminary step length η is given by

$$\eta \equiv \min \left\{ \begin{array}{l} \min\{-g_a^\dashv(\ell_a^{\pi,\delta}) - \pi_i + \pi_j \mid a = i \rightarrow j \in \Delta_A^+ T\} \\ \min\{g_a^\vdash(u_a^{\pi,\delta}) + \pi_i - \pi_j \mid a = i \rightarrow j \in \Delta_A^- T\} \end{array} \right\},$$

and then β is obtained from AdjustFlow.

We start with two technical lemmas showing that key properties of δ-MPC cancellation carry over to the separable convex case.

Lemma 6.2. *Suppose that T is a δ-MPC w.r.t. π, and we cancel T to get node potential π'. If $a \in \Delta^+ T$ $(a \in \Delta^- T)$, then $\ell_a^{\pi,\delta} \leq u_a^{\pi'}$ $(u_a^{\pi,\delta} \geq \ell_a^{\pi'})$. When a is the determining arc, $\ell_a^{\pi,\delta} \geq \ell_a^{\pi'}$ $(u_a^{\pi,\delta} \leq u_a^{\pi'})$.*

Proof. If $a \in \Delta^+ T$, we have $\pi_i < \pi'_i$, $\pi_j = \pi'_j$, and

$$\pi'_i - \pi_i = \beta \leq \pi_j - \pi_i - g_a^\dashv(\ell_a^{\pi,\delta}) \Rightarrow g^\dashv(\ell^{\pi,\delta}) \leq \pi'_j - \pi'_i.$$

It follows from the definition of $u_a^{\pi'}$ that $g_a^\dashv(u_a^{\pi'} + \epsilon) + \pi'_i - \pi'_j > 0$ for any $\epsilon > 0$. Hence it holds that $g_a^\dashv(u_a^{\pi'} + \epsilon) > -\pi'_i + \pi'_j \geq g_a^\dashv(\ell_a^{\pi,\delta})$, which, together with the convexity of g_a, implies $u_a^{\pi'} + \epsilon > \ell_a^{\pi,\delta}$, and thus $u_a^{\pi'} \geq \ell_a^{\pi,\delta}$.

When $a \in \Delta^+ T$ is the determining arc, $-\pi'_i + \pi'_j = g_a^\dashv(\ell_a^{\pi,\delta})$ holds. It follows from the definition of $\ell_a^{\pi'}$ that $g_a^\vdash(\ell_a^{\pi'} - \epsilon) + \pi'_i - \pi'_j < 0$ for any $\epsilon > 0$. Hence we have $g_a^\dashv(\ell_a^{\pi,\delta}) > g_a^\vdash(\ell_a^{\pi'} - \epsilon) \geq g_a^\dashv(\ell_a^{\pi'} - \epsilon)$, which, together with the convexity of g_a, implies $\ell_a^{\pi,\delta} > \ell_a^{\pi'} - \epsilon$, and thus $\ell_a^{\pi,\delta} \geq \ell_a^{\pi'}$. The case $a \in \Delta^- T$ is similar. □

Corollary 6.3. *Suppose that T is a δ-MPC w.r.t. π, and we cancel T to get node potential π'. For a max flow φ from the feasibility algorithm, we have $\ell^{\pi',\delta} \leq \varphi \leq u^{\pi',\delta}$. If a is the determining arc, $\ell_a^{\pi',\delta} + \delta \leq \varphi_a \leq u_a^{\pi',\delta} - \delta$ holds.*

Proof. An arc a with $\ell_a^{\pi,\delta} < \ell_a^{\pi',\delta}$ belongs to $\Delta^- T$, which implies $\varphi_a = u_a^{\pi,\delta} \geq \ell_a^{\pi'} = \ell_a^{\pi',\delta} + \delta$. On the other hand, if $u_a^{\pi,\delta} > u_a^{\pi'\delta}$, the arc a belongs to $\Delta^+ T$, which implies $\varphi_a = \ell_a^{\pi,\delta} \leq u_a^{\pi'} = u_a^{\pi',\delta} - \delta$. Hence, we have $\ell^{\pi',\delta} \leq \varphi \leq u^{\pi',\delta}$. If $a \in \Delta^+ T$ is the determining arc, $\varphi_a \leq u_a^{\pi',\delta} - \delta$ is shown in the previous sentence. From Lemma 6.2, we have $\ell_a^{\pi',\delta} + \delta = \ell_a^{\pi'} \leq \ell_a^{\pi,\delta} \leq \varphi_a$. The case where $a \in \Delta^- T$ is the determining arc is similar. □

It is easy to check that the above lemmas are enough to show that the analogues of Lemma 4.2, Corollary 4.3, and Lemma 4.4 hold for separable convex costs.

When the g_a are general, we must assume that we have an initial φ^0 and bound B such that φ^0 is B-optimal, and the best we can hope for is to compute a solution that is ϵ-optimal for some $\epsilon > 0$ in time polynomial in $\log(B/\epsilon)$. But if the g_a are convex piecewise linear with integral breakpoints, then we can compute an exact optimal solution in polynomial time.

Theorem 6.4. *A scaling phase of δ-MPC canceling cancels at most $m'n$ δ-MPCs. Thus δ-MPC canceling takes $\mathrm{O}(n^3 \log(B/\epsilon)\mathrm{FA}) = \mathrm{O}(n^6 h \log(B/\epsilon))$ time to find an ϵ-optimal solution. When ℓ, u, and f are integer valued and each g_a is a piecewise linear convex function with integral breakpoints, then δ-MPC canceling finds an exact optimal solution in $\mathrm{O}(n^6 \log(nU))$ time.*

Proof. Lemma 4.5 shows that the δ-MPC value of the first cut in a phase is at most $m'\delta$. It takes at most n iterations to reduce this by δ, so there are at most $m'n = \mathrm{O}(n^3)$ iterations per phase. The time per iteration is dominated by computing a δ-MPC, which is O(FA). When ℓ, u, and f are integer valued and each g_a is a piecewise linear function with integral breakpoints, then a proof similar to [5, Lemma 5.1] shows that a δ-optimal π with $\delta < 1/m'$ is exactly optimal, so it suffices to choose $\epsilon = 1/m'$ in this case. □

It is also possible to show, similar to [30], that when each g_a is piecewise linear and quadratic convex, we can compute an exact optimal solution in polynomial time. Similarly, if each g_a is piecewise linear, then we could compute an exact optimal solution in strongly polynomial time.

7. Max mean cut canceling. This section shows that machinery we developed for δ-MPC canceling also leads to a polynomial max mean cut canceling algorithm for SF. We are able to get this algorithm despite the negative result of [35] because we relax f by adding the arc set E, which is outside the class of algorithms considered in [35].

At each iteration, the max mean cut canceling algorithm finds a max mean cut T and calls AdjustFlow with T to get a step length β. We then cancel T by amount β and repeat until no more positive cuts exist, at which point we are optimal, according to Lemma 3.1.

Lemma 7.1. *Suppose we cancel a max mean cut T w.r.t. π with mean value $\delta(\pi)$ to get node potential π'. Then $\overline{V}^{\pi'}(S) \leq \overline{V}^{\pi}(T)$ holds for every cut S.*

Proof. Let φ be a feasible flow in $\widehat{G}^{\pi}$ with data $\ell^{\pi,\delta(\pi)}$, $u^{\pi,\delta(\pi)}$, and f^{π}. Since the step length β is at most η, we have $\ell_a^{\pi'} - \delta(\pi) \leq \varphi_a \leq u_a^{\pi'} + \delta(\pi)$ for every $a \in A$.

As a result of AdjustFlow, we obtain a flow $\psi \in \mathbf{R}^E$ with $0 \leq \psi_e < \delta(\pi)$ for $e \in E$. Put $\varphi'_a = \varphi_a$ for $a \in A$ and $\varphi'_e = \varphi_e - \psi_e$ for $e \in E$. As $\partial_I \varphi \in \mathrm{B}(f^{\pi})$, it follows from the claim in Lemma 4.2 that $x' \equiv \partial_I \varphi' \in \mathrm{B}(f^{\pi'})$. Since $\psi_e > 0$ implies $\varphi_e = \delta(\pi)$, every $e \in E$ satisfies $0 = \ell_e^{\pi'} \leq \varphi'_e \leq \delta(\pi) = u_e^{\pi'} + \delta(\pi)$. Therefore we have

$$
\begin{aligned}
\overline{V}^{\pi'}(S) &= \frac{\ell^{\pi'}(\Delta_A^+ S) - u^{\pi'}(\Delta_A^- S) - f^{\pi'}(S)}{|\Delta_A S| + |S| \cdot |N - S|} && \text{(definition of } \overline{V}^{\pi'}(S)\text{)} \\
&= \frac{\ell^{\pi'}(\Delta_I^+ S) - u^{\pi'}(\Delta_I^- S) - f^{\pi'}(S)}{|\Delta_A^+ S| + |\Delta_I^- S|} && (\ell_e^{\pi'} = u_e^{\pi'} = 0) \\
&\leq \frac{\partial_I \varphi'(S) + \delta(\pi) \cdot |\Delta_A^+ S| + \delta(\pi) \cdot |\Delta_I^- S| - x'(S)}{|\Delta_A^+ S| + |\Delta_I^- S|} && \text{(feasibility of } \varphi'\text{)} \\
&= \delta(\pi) && (\partial_I \varphi'(S) = x'(S)) \\
&= \overline{V}^{\pi}(T) && \text{(definition of } \delta(\pi)\text{)}.
\end{aligned}
$$
□

Corollary 7.2. *With the same hypothesis and notation as Lemma 7.1, if the determining arc $a \in A \cup E$ crosses S, then we have $\overline{V}^{\pi'}(S) \leq (1 - 1/m')\overline{V}^{\pi}(T)$.*

Proof. If the determining arc $a \in A$, then $\ell_a^{\pi'} = \ell_a \leq \varphi_a \leq u_a = u_a^{\pi'}$, and hence $\ell^{\pi'}(\Delta_A^+ S) - u^{\pi'}(\Delta_A^- S) \leq \varphi'(\Delta_A^+ S) + \delta(\pi) \cdot |\Delta_A^+ S| - \varphi'(\Delta_A^- S) + \delta(\pi) \cdot |\Delta_A^- S| - \delta(\pi)$. If $a \in \Delta_E^+ S$, then $\varphi'_a = \delta(\pi) = \ell_a^{\pi'} + \delta(\pi)$, and hence $\ell^{\pi'}(\Delta_E^+ S) \leq \varphi'(\Delta_E^+ S) - \delta(\pi)$. If $a \in \Delta_E^- S$, then, as in the proof of Lemma 4.2, $f^{\pi'}(S) \geq x'(S) + \delta(\pi)$. In all cases we have

$$\begin{aligned}\overline{V}^{\pi'}(S) &\leq \frac{\varphi'(\Delta_I^+ S) + \delta(\pi)\cdot|\Delta_A^+ S| - \varphi'(\Delta_I^- S) + \delta(\pi)\cdot|\Delta_I^- S| - x'(S) - \delta(\pi)}{|\Delta_A^+ S| + |\Delta_I^- S|} \\ &= \delta(\pi) - \frac{\delta(\pi)}{|\Delta_A^+ S| + |\Delta_I^- S|} \leq (1 - 1/m')\overline{V}^{\pi}(T). \qquad \square\end{aligned}$$

Now the analogue of Lemma 4.4 follows just as before.

LEMMA 7.3. *After at most n cut cancellations, the value of a max mean cut decreases by a factor of at most* $(1 - 1/m')$. □

This yields our first bound on the running time of max mean cut canceling.

THEOREM 7.4. *Max mean cut canceling cancels at most* $\mathrm{O}(m'n\log(nU)) = \mathrm{O}(n^3\log(nU))$ *max mean cuts.*

Proof. Let π^0 be the initial node potential. From Lemma 4.1, $\delta(\pi^0) \leq 2U$, and if $\delta(\pi) < 1/m'$ then π is optimal. Hence there are $\mathrm{O}(m'n\log(nU))$ iterations. □

The strongly polynomial argument in section 5 applies to max mean cut canceling as well.

THEOREM 7.5. *Max mean cut canceling cancels at most* $\mathrm{O}(n^5\log n)$ *max mean cuts and takes* $\mathrm{O}(n^{10}h\log n)$ *time.*

Proof. Lemma 7.3 shows that after $m'n\lceil\log(m'+1)\rceil$ cancellations, $\delta(\pi)$ decreases by a factor of at most $1/(m'+1)$. Applying Lemma 5.1 and the proof of Theorem 5.2, we can newly restrict the sign of a $c_a^{\pi^*}$ for some I-arc a every $\mathrm{O}(m'n\log n)$ iterations. Therefore after $\mathrm{O}((m')^2 n\log n) = \mathrm{O}(n^5\log n)$ cancellations, we must be optimal. Since computing a max mean cut requires $\mathrm{O}(n^2\mathrm{FA}) = \mathrm{O}(n^5 h)$ time, we obtain the claimed complexity. □

It seems likely that a faster version of max mean cut canceling could be developed along the lines of the dual cancel and tighten algorithm of [5], but this seems not so interesting in light of the faster algorithm of [7]. Also, it appears to be straightforward to combine the results of this section with the previous section to get a max mean cut canceling algorithm for the separable convex costs case (as is done in [30]), but again this seems not so interesting since the algorithm of section 6 is faster.

8. Crossing submodular functions. We return to the original Edmonds and Giles [4] assumption that f is only crossing submodular on $\mathcal{D}$. Recall that in this case we can efficiently check whether $\mathrm{B}(f) = \emptyset$ by the bi-truncation algorithm of Frank and Tardos [12], and that Fujishige [14] showed that a nonempty base polyhedron $\mathrm{B}(f)$ of a crossing submodular function f on a crossing family $\mathcal{D}$ is identical to the base polyhedron $\mathrm{B}(\widetilde{f})$ of a submodular function $\widetilde{f}$ on a ring family $\widetilde{\mathcal{D}}$ that contains $\mathcal{D}$.

We used the results in the previous paragraph to partially justify that we could implement our algorithms using the ring submodular $\widetilde{f}$ and $\widetilde{\mathcal{D}}$ in place of the crossing submodular f and $\mathcal{D}$. However, it is not yet clear how to compute cut values, exchange capacities, etc. for $\widetilde{f}$ when we are given only an oracle for f.

All of our algorithms are based on applying the feasibility algorithm to various residual networks. The feasibility algorithm needs an initial base to work with. This can be computed using the bi-truncation algorithm. It also needs an oracle for computing exchange capacities $\widetilde{r}(x, i, j)$. Note that $\widetilde{r}(x, i, j)$ is the minimum value of $f(S) - x(S)$ over S containing i but not j. The subfamily of members of $\mathcal{D}$ containing i but not j is a ring family on which f is ring submodular, and so it is reasonable to assume that we have an oracle for computing these exchange capacities.

This will have the effect of simulating the action of the algorithm on $\widetilde{f}$ and $\widetilde{\mathcal{D}}$, meaning that the cut values V^{π} in (3.1) will be computed relative to $\widetilde{f}^{\pi}$ instead of f^{π}, and the feasibility algorithm provides a tight set for $\widetilde{f}^{\pi}$, not for f^{π}. However, since we are always computing exchange capacities w.r.t. f and not $\widetilde{f}$, in fact we never need to compute a value of $\widetilde{f}$ during the algorithm (although the analysis uses $\widetilde{f}$).

Acknowledgments. We heartily thank Lisa Fleischer for many valuable conversations about this paper and helpful comments on previous drafts of it, and an anonymous referee for suggesting several improvements to the paper.

REFERENCES

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows—Theory, Algorithms, and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[2] W. Cui and S. Fujishige, *A primal algorithm for the submodular flow problem with minimum-mean cycle selection*, J. Oper. Res. Soc. Japan, 31 (1988), pp. 431–440.

[3] W. H. Cunningham and A. Frank, *A primal-dual algorithm for submodular flows*, Math. Oper. Res., 10 (1985), pp. 251–262.

[4] J. Edmonds and R. Giles, *A min-max relation for submodular functions on graphs*, Ann. Discrete Math., 1 (1977), pp. 185–204.

[5] T. R. Ervolina and S. T. McCormick, *Two strongly polynomial cut canceling algorithms for minimum cost network flow*, Discrete Appl. Math., 46 (1993), pp. 133–165.

[6] T. R. Ervolina and S. T. McCormick, *Canceling most helpful total cuts for minimum cost network flow*, Networks, 23 (1993), pp. 41–52.

[7] L. Fleischer, S. Iwata, and S. T. McCormick *A faster capacity scaling algorithm for minimum cost submodular flow*, Math. Program., 92 (2002), pp. 119–139.

[8] A. Frank, *A weighted matroid intersection algorithm*, J. Algorithms, 2 (1981), pp. 328–336.

[9] A. Frank, *An algorithm for submodular functions on graphs*, Ann. Discrete Math., 16 (1982), pp. 97–120.

[10] A. Frank, *Finding feasible vectors of Edmonds-Giles polyhedra*, J. Combin. Theory Ser. B, 36 (1984), pp. 221–239.

[11] A. Frank and É. Tardos, *An application of simultaneous Diophantine approximation in combinatorial optimization*, Combinatorica, 7 (1987), pp. 49–65.

[12] A. Frank and É. Tardos, *Generalized polymatroids and submodular flows*, Math. Program., 42 (1988), pp. 489–563.

[13] S. Fujishige, *Algorithms for solving the independent-flow problems*, J. Oper. Res. Soc. Japan, 21 (1978), pp. 189–204.

[14] S. Fujishige, *Structures of polyhedra determined by submodular functions on crossing families*, Math. Program., 29 (1984), pp. 125–141.

[15] S. Fujishige, *Capacity-rounding algorithm for the minimum-cost circulation problem: A dual framework of the Tardos algorithm*, Math. Program., 35 (1986), pp. 298–309.

[16] S. Fujishige, *Submodular Functions and Optimization*, North-Holland, Amsterdam, 1991.

[17] S. Fujishige and X. Zhang, *New algorithms for the intersection problem of submodular systems*, Japan J. Indust. Appl. Math., 9 (1992), pp. 369–382.

[18] S. Fujishige, H. Röck, and U. Zimmermann, *A strongly polynomial algorithm for minimum cost submodular flow problems*, Math. Oper. Res., 14 (1989), pp. 60–69.

[19] A. V. Goldberg and R. E. Tarjan, *A new approach to the maximum flow problem*, J. ACM, 35 (1988), pp. 921–940.

[20] A. V. Goldberg and R. E. Tarjan, *Finding minimum-cost circulations by canceling negative cycles*, J. ACM, 36 (1989), pp. 873–886.

[21] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, New York, 1988.

[22] R. Hassin, *Algorithm for the minimum cost circulation problem based on maximizing the mean improvement*, Oper. Res. Lett., 12 (1992), pp. 227–233.

[23] M. Iri and N. Tomizawa, *An algorithm for finding an optimal "independent assignment,"* J. Oper. Res. Soc. Japan, 19 (1976), pp. 32–57.

[24] S. Iwata, *A capacity scaling algorithm for convex cost submodular flows*, Math. Program., 76 (1997), pp. 299–308.

[25] S. Iwata, L. Fleischer, and S. Fujishige, *A combinatorial strongly polynomial algorithm for minimizing submodular functions*, J. ACM, 48 (2001), pp. 761–777.

[26] S. Iwata, S. T. McCormick, and M. Shigeno, *A strongly polynomial cut canceling algorithm for the submodular flow problem*, in Integer Programming and Combinatorial Optimization, Proceedings of the Seventh IPCO Conference, G. Cornuéjols, R. E. Burkard, and G. J. Woeginger, eds., Springer, Berlin, 1999, pp. 259–272.

[27] S. Iwata, S. T. McCormick, and M. Shigeno, *A fast cost scaling algorithm for submodular flow*, Inform. Process. Lett., 74 (2000), pp. 123–128.

[28] S. Iwata, S. T. McCormick, and M. Shigeno, *Fast cycle canceling algorithms for minimum cost submodular flow*, Combinatorica, 23 (2003), pp. 503–525.

[29] S. Iwata, S. T. McCormick, and M. Shigeno, *A Relaxed Cycle-Canceling Approach to Separable Convex Optimization in Unimodular Linear Space*, manuscript.

[30] A. V. Karzanov and S. T. McCormick, *Polynomial methods for separable convex optimization in unimodular linear spaces with applications*, SIAM J. Comput., 26 (1997), pp. 1245–1275.

[31] E. L. Lawler and C. U. Martel, *Computing maximal polymatroidal network flows*, Math. Oper. Res., 7 (1982), pp. 334–347.

[32] S. T. McCormick, *Submodular function minimization*, in Handbook on Discrete Optimization, K. Aardal, G. Nemhauser, and R. Weismantel, eds., Elsevier, to appear.

[33] S. T. McCormick and T. R. Ervolina, *Cancelling most helpful total submodular cuts for submodular flow*, in Integer Programming and Combinatorial Optimization (Proceedings of the Third IPCO Conference), G. Rinaldi and L. A. Wolsey eds., 1993, pp. 343–353.

[34] S. T. McCormick and T. R. Ervolina, *Computing maximum mean cuts*, Discrete Appl. Math., 52 (1994), pp. 53–70.

[35] S. T. McCormick, T. R. Ervolina, and B. Zhou, *Mean Canceling Algorithms for General Linear Programs and Why They (Probably) Don't Work for Submodular Flow*, UBC Faculty of Commerce Working Paper 94-MSC-011, Vancouver, BC, Canada, 1994.

[36] S. T. McCormick and A. Shioura, *Minimum ratio canceling is polynomial for linear programming, but not strongly polynomial, even for networks*, Oper. Res. Lett., 27 (2000), pp. 199–207.

[37] T. Radzik, *Newton's method for fractional combinatorial optimization*, in Proceedings of the 33rd IEEE Annual Symposium on Foundations of Computer Science, 1992, pp. 659–669.

[38] T. Radzik, *Parametric flows, weighted means of cuts, and fractional combinatorial optimization*, in Complexity in Numerical Optimization, P. Pardalos, ed., World Scientific, River Edge, NJ, 1993, pp. 351–386.

[39] P. Schönsleben, *Ganzzahlige Polymatroid-Intersektions Algorithmen*, Dissertation, ETH Zürich, 1980.

[40] A. Schrijver, *Total dual integrality from directed graphs, crossing families, and sub- and supermodular functions*, in Progress in Combinatorial Optimization, W. R. Pulleyblank, ed., Academic Press, New York, 1984, pp. 315–361.

[41] A. Schrijver, *A combinatorial algorithm minimizing submodular functions in strongly polynomial time*, J. Combin. Theory Ser. B, 80 (2000), pp. 346–355.

[42] M. Shigeno, S. Iwata, and S. T. McCormick, *Relaxed most negative cycle and most positive cut canceling algorithms for minimum cost flow*, Math. Oper. Res., 25 (2000), pp. 76–104.

[43] É. Tardos, *A strongly polynomial minimum cost circulation algorithm*, Combinatorica, 5 (1985), pp. 247–255.

[44] C. Wallacher and U. Zimmermann, *A polynomial cycle canceling algorithm for submodular flows*, Math. Program., 86 (1999), pp. 1–15.

[45] U. Zimmermann, *Negative circuits for flows and submodular flows*, Discrete Appl. Math., 36 (1992), pp. 179–189.

SIAM J. Discrete Math.
Vol. 19, No. 2, pp. 321–344

STABILIZATION OF BLOCK-TYPE-DECODABILITY PROPERTIES FOR CONSTRAINED SYSTEMS*

PANU CHAICHANAVONG[†] and BRIAN H. MARCUS[‡]

Abstract. We consider a class of encoders for constrained systems, which we call block-type-decodable encoders. For a constrained system presented by a deterministic graph, we design a block-type-decodable encoder by selecting a subset of states of the graph to be used as encoder states. Such a subset is known as a set of principal states. Our goal is to find an optimal set of principal states, i.e., a set which yields the highest code rate. We study the relationship between optimal sets of principal states at finite block length and at asymptotically large block length. Specifically, we show that for a primitive constraint and a large enough block length, any optimal set of principal states is also asymptotically optimal. Moreover, we give bounds on the block length such that this relationship holds. We also characterize asymptotically optimal block-type-decodable encoders. Finally, we study the complexity of various problems related to block-type-decodable encoders.

Key words. constrained systems, block encoders, block-decodable encoders, deterministic encoders, sets of principal states, integer programming, NP-complete problems

AMS subject classifications. 94A99, 94B99, 68R10, 90C90, 68Q17

DOI. 10.1137/S0895480104443679

1. Introduction.

In most recording channels, arbitrary data is encoded into constrained sequences to improve the performance of storage systems. A constraint is presented by a labeled finite directed graph, and a constrained sequence is obtained by reading the labels of a path in the graph. The best known constraint is the runlength-limited (RLL(d,k)) constraint, which is the binary constraint that bounds the lengths of the runs of zeros to be at least d and at most k (see Figure 1.1). This constraint is used in magnetic tape drives and optical drives to suppress the interference between adjacent bits and improve the timing recovery system. The constraint that we will use as an example throughout this paper is the asymmetric-RLL(d_0, k_0, d_1, k_1) (see, e.g., Immink [7, section 4.5]), which requires that the lengths of the runs of zeros are between d_0 and k_0 and the lengths of the runs of ones are between d_1 and k_1.

For a given constraint and a given block length q, we consider fixed-rate encoders that encode arbitrary user data into constrained blocks of length q such that strings formed from concatenating consecutive encoded blocks satisfy the constraint. The precise definitions of the encoders that we consider in this paper are given in sections 2 and 3.

In order to avoid error propagation in the decoding process, many practical applications use block encoders. Although these encoders are conceptually simplest, we may be able to achieve higher rates using block-decodable encoders for which error propagation is still limited to one block. However, the optimal rate is difficult to compute, and an achieving block-decodable encoder is hard to design. Nevertheless, for some constraints—including the RLL(d,k) constraint—this problem has been shown

*Received by the editors May 12, 2004; accepted for publication (in revised form) November 29, 2004; published electronically October 20, 2005. Part of this work was presented at the 2003 IEEE International Symposium on Information Theory, Yokohama, Japan.

http://www.siam.org/journals/sidma/19-2/44367.html

[†] Center for Magnetic Recording Research, University of California-San Diego, 9500 Gilman Drive, La Jolla, CA 92093 (panu@ucsd.edu).

[‡] Department of Mathematics, University of British Columbia, 1984 Mathematics Road, Vancouver V6T 1Z2, BC, Canada (marcus@math.ubc.ca).

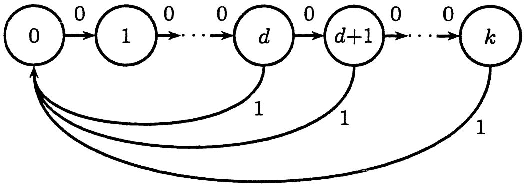


FIG. 1.1. *Presentation of RLL(d, k) constraint.*

to be equivalent to the problem of designing a deterministic encoder [4], which is much more tractable.

In this work, we are interested in these three classes of encoders, which we call block-type-decodable encoders: block, block-decodable, and deterministic encoders. It is known that the characterization of block-type-decodable encoders can be specified by subsets of states called the sets of principal states [10].

An optimal set of principal states for a deterministic encoder can be found using the Franaszek algorithm [4]. Algorithms for computing the optimal sets of principal states for a block encoder were presented by Freiman and Wyner [5] and Marcus, Siegel, and Wolf [11]; in this paper, we present a new framework for this problem. We also give candidates for optimal sets of principal states for a block-decodable encoder, together with upper and lower bounds on the optimal code rate; this is based on an integer programming interpretation.

Typically, high code rates require large block lengths. Thus, it is of interest to study the relationship between the optimal sets of principal states at a finite block length and those at asymptotically large block length. In [3], for deterministic encoders, we showed how to compute an asymptotically optimal set of principal states and observed that this is sometimes easier than the same problem at a finite block length. Empirically, this asymptotically optimal set of principal states is a good approximation to the finite case. In the present paper, we show how to compute an asymptotically optimal set of principal states for block and block-decodable encoders. We will establish the relationship between the finite case and the asymptotic case by showing that for a primitive constraint, there is a q_0 such that for any $q \geq q_0$, any optimal set of principal states at block length q is also asymptotically optimal. An upper bound on q_0 is given for each class of encoder; empirically, this bound appears to be small.

Finally, we consider the complexity of designing optimal block-type-decodable encoders. For deterministic encoders, this is known to be polynomial because the Franaszek algorithm is polynomial. Ashley, Karabed, and Siegel [1] showed that the problem of designing block-decodable encoders is NP-complete. In section 8, we show that the complexity of designing a block encoder is also NP-complete. We further show that if the number of states is fixed, all of these problems can be solved in polynomial time.

2. Background.

Here we summarize basic definitions in constrained coding used in this paper. More detail can be found in [10, 7].

A *labeled directed graph* (or simply a *graph*) $G = (V, E, L)$ consists of a finite set of states $V = V_G$, a finite set of edges $E = E_G$ where each edge has an initial state and terminal state in V_G, and an edge labeling $L = L_G : E \to \Sigma$ where Σ is a finite alphabet. We will sometimes refer to a label or a sequence of labels of G as a *word.*

A *constrained system* or *constraint* $S = S(G)$ is the set of finite sequences obtained by reading the edge labels of a path in a labeled graph G. Such a graph is called a *presentation* of the constraint.

Two important properties of a graph are irreducibility and primitivity. A graph is *irreducible* if for any given pair u, v of states, there is a path from u to v and a path from v to u. A graph that is not irreducible is called *reducible*. Such a graph consists of nonoverlapping irreducible subgraphs, called *irreducible components*, and transitional edges between them. A graph is *primitive* if there exists a positive integer ℓ such that for all pairs u, v of states, there are paths from u to v and v to u of length ℓ. A constrained system is said to be irreducible if it has an irreducible presentation. Similarly, a constrained system is primitive if it has a primitive presentation. From the definitions, we can see that primitivity is stronger in the sense that every primitive graph (constrained system) is irreducible. Many practical constraints including RLL(d, k) are primitive.

Irreducibility and primitivity are properties of the topology of a graph alone but not its labeling. We now state the definitions of important properties of graph labeling that are used throughout the paper.

- A labeled graph is *deterministic* if at each state, all outgoing edges carry distinct labels. It is well known that every constraint has a deterministic presentation. Furthermore, for an irreducible constraint, there is a unique minimal (in terms of the number of states) deterministic presentation, called the *Shannon cover*. This presentation is often used as a starting point to construct a constrained encoder.
- A labeled graph has *finite memory* if there is an integer N such that all paths of length N with the same labeling terminate at the same state. The smallest N for which this holds is called the *memory* of the graph.
- A labeled graph is *lossless* if any two distinct paths with the same initial state and terminal state have different labels. This is the weakest property among all mentioned properties of labeling.

Let G be a labeled graph. The *adjacency matrix* $A = A_G$ is the $|V_G| \times |V_G|$ matrix whose entry $A_{u,v}$ is the number of edges from state u to state v in G. We say that a matrix is irreducible if it is the adjacency matrix of an irreducible graph. Similarly, a matrix is primitive if it is the adjacency matrix of a primitive graph.

Let G be a labeled graph. The *qth power of* G, denoted G^q, is the labeled graph with the same set of states as G, but with one edge for each path of length q in G, labeled by the word of length q generated by that path. For a constraint S presented by a labeled graph G, the *qth power of* S, denoted S^q, is the constraint presented by G^q. If A is the adjacency matrix of G, it can be shown that the adjacency matrix of G^q is A^q.

The *capacity* of a constraint S, denoted $\mathrm{cap}(S)$, is defined to be

$$\mathrm{cap}(S) = \lim_{q \to \infty} \frac{1}{q} \log N(q; S),$$

where $N(q; S)$ is the number of words of length q in S. (In this paper, the logarithmic function has base 2.) The capacity measures the growth rate of the number of words in S. It is known that $\mathrm{cap}(S^q) = q\mathrm{cap}(S)$.

To express the capacity in terms of the adjacency matrix of a lossless (in particular, deterministic) presentation G of S, we need the following notation. For a square matrix A, we denote by $\lambda(A)$ the *spectral radius* of A, that is, the largest of the absolute values of the eigenvalues of A. According to the Perron–Frobenius theory [12], $\lambda(A)$ is an eigenvalue of A. It is well known that

$$\mathrm{cap}(S) = \log \lambda(A_G).$$

Let S be a constrained system and let n be a positive integer. An (S, n) *encoder* is a labeled graph $\mathcal{E}$ such that

- each state of $\mathcal{E}$ has *out-degree* n, i.e., n outgoing edges,
- $S(\mathcal{E}) \subseteq S$,
- $\mathcal{E}$ is lossless.

The labels of the encoder are sometimes called *output labels.* A *tagged* (S, n) *encoder* is an (S, n) encoder whose outgoing edges from each state are assigned distinct *input tags* from an alphabet of size n, and this defines an encoding function. For an (S^q, n) encoder, we define the *block length* to be q and the *rate* to be $(\log n)/q$. It is known that $\mathrm{cap}(S)$ is an upper bound on the rate of any (S^q, n) encoder.

3. Block-type-decodable encoders. In this paper, we restrict our interest to block, block-decodable, and deterministic encoders. A *block encoder* (blk) is a finite-state encoder such that any two edges have the same input tag if and only if they have the same output label. A *block-decodable encoder* (blkdec) is a finite-state encoder such that any two edges with the same output label have the same input tag. A *deterministic encoder* (det) is a finite-state encoder with deterministic output labeling.

It is easy to see that a block encoder is block decodable, which in turn is deterministic. A block-decodable encoder can be viewed as a deterministic encoder with a consistent input tag assignment. In this paper, we focus on these three classes of encoders which we call *block-type-decodable encoders.*

For a constrained system S, a class of encoders $\mathcal{C} \in \{\mathrm{blk}, \mathrm{blkdec}, \mathrm{det}\}$, and a block length q, define $M_{\mathcal{C}}(q)$ to be the maximum n such that there exists an (S^q, n) encoder in class $\mathcal{C}$. Suppose that S is irreducible and let G be an irreducible deterministic presentation of S. For each class $\mathcal{C}$ of block-type-decodable encoders, it can be shown that there exists an (S, n) encoder in class $\mathcal{C}$ if and only if there exists such an encoder which is a subgraph of G. (For block encoder, see [5]. For block-decodable encoder, this is a special case of [2, Corollary 12.2]. For deterministic encoder, see [4]. For a unified treatment, see [10].) Thus the problem of designing block-type-decodable encoders can be solved by choosing a subgraph of G. This can be broken into two steps: First, choose a set of states, called a *set of principal states.* (A principal set of states may be a more appropriate term, but we will follow Franaszek [4] who defined it for deterministic encoders.) Then choose edges.

The reason for breaking this into two steps is that we often need to design an (S^q, n) encoder for various block lengths q. Since the graphs G^q have the same set of states for all q, we may need to solve the first step only once. In fact, the problem of determining whether a set of principal states is optimal for all large enough q is one of the main themes of the paper. That is, we study whether the optimal sets of principal states stabilize and, if so, at what value of q. (A set of principal states is optimal if it induces an encoder with the highest rate. For a more precise definition, see below.)

Let $M_{\mathcal{C}}(q, P)$ denote the maximum n such that there exists an (S^q, n) encoder in class $\mathcal{C}$ constructed from the set of principal states P. Therefore we can write $M_{\mathcal{C}}(q) = \max_{P \subseteq V_G} M_{\mathcal{C}}(q, P)$. Moreover, we say that P *achieves* $M_{\mathcal{C}}(q)$ if $M_{\mathcal{C}}(q, P) = M_{\mathcal{C}}(q)$. We shall later refer to such a set P as an *optimal set of principal states.*

In order to quantify the optimality of block-type-decodable encoders, we need the following notations. Let u and v be any states in a labeled graph G. The *follower set* of u in G, denoted $\mathcal{F}_G(u)$, is the set of all finite words that can be generated from u in G. We shall use $\mathcal{F}_G^q(u, v)$ to denote the set of all words of length q that can be generated in G by paths that start at u and terminate at v. Similarly, for a set of states P, $\mathcal{F}_G^q(u, P)$ denotes the set of all words of length q that can be generated in G by paths that start at u and terminate at a state in P, i.e., $\mathcal{F}_G^q(u, P) = \bigcup_{v \in P} \mathcal{F}_G^q(u, v)$.

The states of a labeled graph are naturally endowed with the *partial ordering* by inclusion of follower sets: $u \preceq v$ if $\mathcal{F}_G(u) \subseteq \mathcal{F}_G(v)$. We say that a set $P \subseteq V_G$ is *complete* if whenever u is in P and $u \preceq v$, then v is also in P.

Based on these notations, Freiman and Wyner [5] showed that

$$M_{\mathrm{blk}}(q) = \max_{P \subseteq V_G} \left| \bigcap_{u \in P} \mathcal{F}_G^q(u, P) \right| .$$

To simplify the search for an optimal P, they further proved that when G has finite memory less than or equal to q, it suffices to consider sets P which are complete. In fact, the following proposition shows that this is true for all classes of block-type-decodable encoders even when the condition that q is greater than the memory is removed.

PROPOSITION 3.1. *Let S be a constrained system with a deterministic presentation G. Let $P \subseteq V_G$ and let P' be the smallest complete set such that $P \subseteq P'$. Then for each class $\mathcal{C}$ of encoder and block length q, $M_{\mathcal{C}}(q, P) \leq M_{\mathcal{C}}(q, P')$.*

Proof. Let $v \in P'$. It suffices to show that there is a state $u \in P$ such that $\mathcal{F}_G^q(u, P) \subseteq \mathcal{F}_G^q(v, P')$.

Since P' is the smallest complete set such that $P \subseteq P'$, there must be a state $u \in P$ such that $u \preceq v$. Let $w \in \mathcal{F}_G^q(u, P)$. Since $u \preceq v$, v can also generate w. Since G is deterministic, the outgoing edges from u and v labeled by w are unique. Denote the terminal states of these edges by $\bar{u}$ and $\bar{v}$, respectively. Then $\bar{u} \preceq \bar{v}$ because $u \preceq v$. Hence $\bar{v} \in P'$ because P' is complete and $\bar{u} \in P \subseteq P'$. This implies that $w \in \mathcal{F}_G^q(v, P')$. ☐

Similar expressions for $M_{\mathrm{det}}(q, P)$ and $M_{\mathrm{det}}(q)$ are due to Franaszek [4]:

$$M_{\mathrm{det}}(q, P) = \min_{u \in P} |\mathcal{F}_G^q(u, P)| = \min_{u \in P} \sum_{v \in P} (A_G^q)_{u,v}, \tag{3.1}$$

$$M_{\mathrm{det}}(q) = \max_{P \subseteq V_G} \min_{u \in P} \sum_{v \in P} (A_G^q)_{u,v}.$$

We do not know of a formula for $M_{\mathrm{blkdec}}(q)$ as simple as those above, but, as with $M_{\mathrm{blk}}(q)$ and $M_{\mathrm{det}}(q)$, it is a function of only an arbitrary irreducible deterministic presentation of the constraint, such as the Shannon cover.

4. Stabilization at large block length. We know from the previous section that to design a block-type-decodable encoder, we need to choose a set of principal states. Our goal is to find an optimal set of principal states that maximizes the code rate. In some cases, it is easier to find such an optimal set of principal states at asymptotically large block length. Thus it is desirable if we can relate the optimal sets of principal states at asymptotically large block length to the ones at finite block length. In this section, we study the relationship between the two.

Recall that for a constraint S with the Shannon cover G, $\mathrm{cap}(S) = \log \lambda(A_G)$. When it is clear from the context, we also denote $\lambda(A_G)$ by λ. From the expression for cap(S), we would expect $M_{\mathcal{C}}(q,P)$ to grow as λ^q. Thus it is natural to define $M^q_{\mathcal{C}}(P) = M_{\mathcal{C}}(q,P)/\lambda^q$. Let $M^\infty_{\mathcal{C}}(P) = \lim_{q\to\infty} M^q_{\mathcal{C}}(P)$. In [3, Proposition 3], we showed that $M^\infty_{\mathrm{det}}(P)$ exists for primitive constraints. We shall prove that $M^\infty_{\mathrm{blk}}(P)$ and $M^\infty_{\mathrm{blkdec}}(P)$ exist for primitive constraints in sections 6 and 7, respectively. We define $M^\infty_{\mathcal{C}} = \max_{P\subseteq V_G} M^\infty_{\mathcal{C}}(P)$. We say that a set P is *asymptotically optimal* if $M^\infty_{\mathcal{C}}(P) = M^\infty_{\mathcal{C}}$. Furthermore, define $\mathcal{P}_{\mathcal{C}}(q)$ and $\mathcal{P}^\infty_{\mathcal{C}}$ to be the collection of optimal sets of principal states at block length q and the collection of asymptotically optimal sets of principal states, respectively. Lastly we define $M^*_{\mathcal{C}} = \lim_{q\to\infty} M_{\mathcal{C}}(q)/\lambda^q$.

Proposition 4.1. *For any class $\mathcal{C}$, if $M^\infty_{\mathcal{C}}(P)$ exists for each $P \subseteq V_G$, then the following hold:*

(i) *$\mathcal{P}_{\mathcal{C}}(q) \subseteq \mathcal{P}^\infty_{\mathcal{C}}$ for sufficiently large q.*

(ii) *$M^*_{\mathcal{C}}$ exists and is equal to $M^\infty_{\mathcal{C}}$.*

A proof of Proposition 4.1 is given later in this section. A slightly different version of this proposition for deterministic encoders appears in [3].

Assuming that the condition in Proposition 4.1 is satisfied, it is natural to wonder when (i) holds. In later sections, we give bounds on q such that this holds for each class of encoder. In order to establish those bounds and to prove Proposition 4.1, we need the following lemma. First, define

$$\epsilon_{\mathcal{C}} = M^\infty_{\mathcal{C}} - \max_{P\notin\mathcal{P}^\infty_{\mathcal{C}}} M^\infty_{\mathcal{C}}(P).$$

Lemma 4.2. *If q satisfies*

$$|M^q_{\mathcal{C}}(P) - M^\infty_{\mathcal{C}}(P)| < \frac{\epsilon_{\mathcal{C}}}{2} \tag{4.1}$$

for each $P \subseteq V_G$, then $\mathcal{P}_{\mathcal{C}}(q) \subseteq \mathcal{P}^\infty_{\mathcal{C}}$.

Proof. Let $P \in \mathcal{P}_{\mathcal{C}}(q)$ and $P^* \in \mathcal{P}^\infty_{\mathcal{C}}$. It follows from (4.1) that

$$M^\infty_{\mathcal{C}}(P) + \frac{\epsilon_{\mathcal{C}}}{2} > M^q_{\mathcal{C}}(P) \geq M^q_{\mathcal{C}}(P^*) > M^\infty_{\mathcal{C}}(P^*) - \frac{\epsilon_{\mathcal{C}}}{2} = M^\infty_{\mathcal{C}} - \frac{\epsilon_{\mathcal{C}}}{2}.$$

Therefore, $M^\infty_{\mathcal{C}} - M^\infty_{\mathcal{C}}(P) < \epsilon_{\mathcal{C}}$, and so $P \in \mathcal{P}^\infty_{\mathcal{C}}$ by the definition of $\epsilon_{\mathcal{C}}$. ∎

In the case that $\mathcal{P}^\infty_{\mathcal{C}}$ has only one element, the condition in the lemma implies that $\mathcal{P}_{\mathcal{C}}(q) = \mathcal{P}^\infty_{\mathcal{C}}$. This allows us to determine the optimal set of principal states at large block length, in particular the block length that satisfies the bounds given in later sections, from the asymptotically optimal set of principal states.

Proof of Proposition 4.1. Suppose that $M^\infty_{\mathcal{C}}(P)$ exists for each $P \subseteq V_G$. Then (4.1) holds for sufficiently large q, and (i) follows by Lemma 4.2.

Since $M^q_{\mathcal{C}}(P)$ is a convergent sequence for each $P \subseteq V_G$,

$$M^*_{\mathcal{C}} = \lim_{q\to\infty} \max_{P\subseteq V_G} M^q_{\mathcal{C}}(P) = \max_{P\subseteq V_G} \lim_{q\to\infty} M^q_{\mathcal{C}}(P) = M^\infty_{\mathcal{C}}.$$

This proves (ii). ∎

5. Stabilization for deterministic encoders. In this section, we study bounds on q such that $\mathcal{P}_{\mathrm{det}}(q) \subseteq \mathcal{P}^\infty_{\mathrm{det}}$ by utilizing the Perron–Frobenius theory [12]. From the Perron–Frobenius theory, an irreducible matrix A has a unique largest positive eigenvalue $\lambda = \lambda(A)$. Moreover, the corresponding right and left eigenvectors, $\mathbf{r}$ and $\mathbf{l}$, have all positive entries. In our context, $\mathbf{r}$ is a column vector and $\mathbf{l}$ is a row vector. Suppose $\mathbf{r}$ and $\mathbf{l}$ are normalized so that $\mathbf{lr} = 1$. Define $\Lambda = \mathbf{rl}$, a rank-one matrix. If A is primitive, then it follows from the Perron–Frobenius theory that

$$\lim_{q\to\infty} \frac{A^q}{\lambda^q} = \Lambda. \tag{5.1}$$

The following result, which gives a characterization of M^*_{det}, is a consequence of [3, Proposition 4] and Proposition 4.1 above.

PROPOSITION 5.1 (see [3]). *For each $P \subseteq V_G$, $M^\infty_{\text{det}}(P)$ exists. Moreover,*

(i) *$\mathcal{P}_{\text{det}}(q) \subseteq \mathcal{P}^\infty_{\text{det}}$ for sufficiently large q,*

(ii) *M^*_{det} exists and is equal to M^∞_{det}.*

Before stating the main result of this section, we provide the definition of the maximum row sum matrix norm. Let A be an $n \times n$ matrix over the complex numbers. Then $\|A\|_\infty$ is defined as

$$\|A\|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} |A_{i,j}|.$$

It can also be written as

$$\|A\|_\infty = \max_{\|x\|_\infty = 1} \|Ax\|_\infty.$$

The following theorem provides a bound on block length q such that an optimal set of principal states for a deterministic encoder is also asymptotically optimal.

THEOREM 5.2. *If q satisfies*

$$\left\| \frac{A^q}{\lambda^q} - \Lambda \right\|_\infty < \frac{\epsilon_{\text{det}}}{2}, \tag{5.2}$$

then $\mathcal{P}_{\text{det}}(q) \subseteq \mathcal{P}^\infty_{\text{det}}$.

Proof. We shall show that if (5.2) holds, then for each P,

$$|M^q_{\text{det}}(P) - M^\infty_{\text{det}}(P)| < \frac{\epsilon_{\text{det}}}{2},$$

and the theorem follows from Lemma 4.2.

Let $\mathbf{x} = (x_u)$ be the *characteristic vector* of P, that is, a 0-1 vector of dimension $|V_G|$ such that $x_u = 1$ if $u \in P$ and $x_u = 0$ otherwise. Define

$$\begin{aligned} \mathbf{y} &= \frac{A^q}{\lambda^q}\mathbf{x}, \\ \mathbf{z} &= \Lambda \mathbf{x}. \end{aligned}$$

From (3.1) and (5.1), one can show that [3]

$$M^q_{\text{det}}(P) = \min_{u \in P} y_u \tag{5.3}$$

and

$$M^\infty_{\text{det}}(P) = \min_{u \in P} z_u. \tag{5.4}$$

Let u and v be states achieving the minimum in (5.3) and (5.4), respectively. Then $y_u = M^q_{\text{det}}(P)$ and $z_v = M^\infty_{\text{det}}(P)$. Furthermore, $y_u \le y_v$ and $z_v \le z_u$. Since $\mathbf{x}$ is a 0-1 vector, $\|\mathbf{x}\|_\infty = 1$. Then it follows from (5.2) that

$$\|\mathbf{y} - \mathbf{z}\|_\infty = \left\| \frac{A^q}{\lambda^q}\mathbf{x} - \Lambda \mathbf{x} \right\|_\infty < \frac{\epsilon_{\text{det}}}{2}.$$

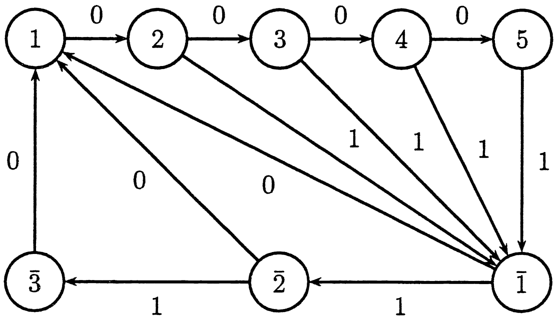


FIG. 5.1. *Shannon cover of the asymmetric-RLL*$(2,5,1,3)$.

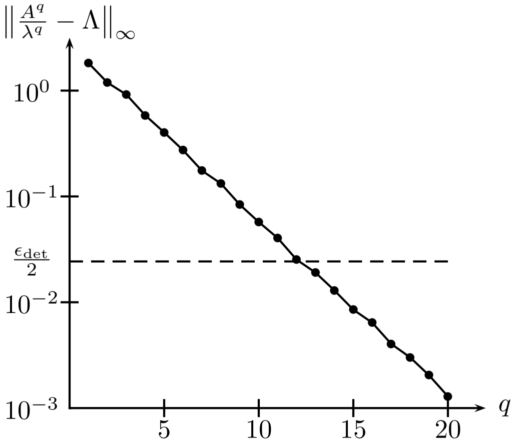


FIG. 5.2. $\|\frac{A^q}{\lambda^q} - \Lambda\|_\infty$.

This implies that $|y_u - z_u| < \epsilon_{\rm det}/2$ and $|y_v - z_v| < \epsilon_{\rm det}/2$. We want to show that $|y_u - z_v| < \epsilon_{\rm det}/2$.

Case 1. Suppose $y_u - z_v \geq \epsilon_{\rm det}/2$. Since $y_u \leq y_v$, we have $y_v - z_v \geq \epsilon_{\rm det}/2$, a contradiction.

Case 2. Suppose $y_u - z_v \leq -\epsilon_{\rm det}/2$. Since $z_v \leq z_u$, we have $y_u - z_u \leq -\epsilon_{\rm det}/2$, a contradiction.

Thus we conclude that $|M^q_{\rm det}(P) - M^\infty_{\rm det}(P)| < \epsilon_{\rm det}/2$. □

Example 5.3. The Shannon cover for the asymmetric-RLL$(2,5,1,3)$ constraint is shown in Figure 5.1.

In contrast to RLL constraints [8, 6], there is no known explicit characterization of the optimal sets of principal states for the asymmetric-RLL constraint. However, we can numerically compute $M_{\rm det}(q)$, $M^*_{\rm det}$, and the achieving set of principal states easily. We obtain $M^*_{\rm det} = 0.7563$, $\epsilon_{\rm det} = 0.0487$, and $P^*_{\rm det} = \{1,2,3,4,\bar{1},\bar{2}\}$ is the only asymptotically optimal set of principal states.

We compute $\|\frac{A^q}{\lambda^q} - \Lambda\|_\infty$ explicitly for small values of q in Figure 5.2. The plot suggests that $\|\frac{A^q}{\lambda^q} - \Lambda\|_\infty < \epsilon_{\rm det}/2$ holds for $q \geq 13$. Since we do not know whether $\|\frac{A^q}{\lambda^q} - \Lambda\|_\infty$ is decreasing with q, we will compute an upper bound for $\|\frac{A^q}{\lambda^q} - \Lambda\|_\infty$.

In this example, A is diagonalizable: $A = TDT^{-1}$, where $D = \operatorname{diag}[\lambda_i]$ is a diagonal matrix with $\lambda = |\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_8|$. Moreover, the first column of T and the first row of T^{-1} are, respectively, the right ($\mathbf{r}$) and left ($\mathbf{l}$) eigenvectors of A associated with the eigenvalue λ normalized so that $\mathbf{lr} = 1$. Then we have

$$\begin{aligned}
\left\| \frac{A^q}{\lambda^q} - \Lambda \right\|_\infty &= \left\| \frac{1}{\lambda^q} TD^qT^{-1} - \mathbf{rl} \right\|_\infty \\
&= \left\| \frac{1}{\lambda^q} T \begin{bmatrix} 0 & & & \\ & \lambda_2^q & & \\ & & \ddots & \\ & & & \lambda_8^q \end{bmatrix} T^{-1} \right\|_\infty \\
&\leq \frac{1}{\lambda^q} \|T\|_\infty \left\| \begin{bmatrix} 0 & & & \\ & \lambda_2^q & & \\ & & \ddots & \\ & & & \lambda_8^q \end{bmatrix} \right\|_\infty \|T^{-1}\|_\infty \\
&= \|T\|_\infty \|T^{-1}\|_\infty \left(\frac{|\lambda_2|}{\lambda} \right)^q \\
&= (2.8811)(3.8981) \left(\frac{1.1271}{1.6372} \right)^q .
\end{aligned}$$

If $q \geq 17$, then

$$\left\| \frac{A^q}{\lambda^q} - \Lambda \right\|_\infty \leq 11.2308(0.6884)^{17} = 0.0197 < 0.0243 = \frac{\epsilon_{\text{det}}}{2}.$$

Therefore P^*_{det} is the only optimal set of principal states for $q \geq 17$.

In fact, by computing $M_{\text{det}}(q, P)$ for $1 \leq q \leq 12$, one can show that P^*_{det} is optimal for all q and is the only optimal set of principal states precisely when $q = 5$ and $q \geq 7$.

With the motivation from the above example, we offer the following corollary.

COROLLARY 5.4. *Let λ_i be the distinct eigenvalues of A with $\lambda_1 = \lambda = \lambda(A)$. Let s_i be the multiplicity of λ_i. Let T be a transformation matrix which decomposes A into Jordan canonical form. If*

$$\frac{1}{\lambda^q} \|T\|_\infty \|T^{-1}\|_\infty \max_{i \geq 2} \sum_{k=0}^{s_i - 1} \binom{q}{k} |\lambda_i|^{q-k} < \frac{\epsilon_{\text{det}}}{2},$$

then $\mathcal{P}_{\text{det}}(q) \subseteq \mathcal{P}^\infty_{\text{det}}$.

Proof. Let J be the Jordan form of all eigenvalues of A other than λ; then

$$\begin{aligned}
\left\| \frac{A^q}{\lambda^q} - \Lambda \right\|_\infty &= \left\| \frac{1}{\lambda^q} T \begin{bmatrix} \lambda^q & 0 \\ 0 & J^q \end{bmatrix} T^{-1} - \Lambda \right\|_\infty \\
&= \left\| \frac{1}{\lambda^q} T \begin{bmatrix} 0 & 0 \\ 0 & J^q \end{bmatrix} T^{-1} \right\|_\infty \\
&\leq \frac{1}{\lambda^q} \|T\|_\infty \|T^{-1}\|_\infty \|J^q\|_\infty \\
&= \frac{1}{\lambda^q} \|T\|_\infty \|T^{-1}\|_\infty \max_{i \geq 2} \|J_i^q\|_\infty, \qquad (5.5)
\end{aligned}$$

where J_i is the Jordan (sub)matrix associated with λ_i.

The Jordan matrix J_i can have several forms. The one which yields the largest $\|J_i^q\|_\infty$ is the one with single block

$$J_i = \begin{bmatrix} \lambda_i & 1 & 0 & \cdots & 0 \\ 0 & \lambda_i & 1 & \cdots & 0 \\ 0 & 0 & \lambda_i & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_i \end{bmatrix}_{s_i \times s_i} .$$

One can show that

$$J_i^q = \begin{bmatrix} \lambda_i^q & \binom{q}{1}\lambda_i^{q-1} & \binom{q}{2}\lambda_i^{q-2} & \cdots & \binom{q}{s_i-1}\lambda_i^{q-s_i+1} \\ 0 & \lambda_i^q & \binom{q}{1}\lambda_i^{q-1} & \cdots & \binom{q}{s_i-2}\lambda_i^{q-s_i+2} \\ 0 & 0 & \lambda_i^q & \cdots & \binom{q}{s_i-3}\lambda_i^{q-s_i+3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_i^q \end{bmatrix} .$$

Therefore

$$\|J_i^q\|_\infty = \sum_{k=0}^{s_i-1} \binom{q}{k} |\lambda_i|^{q-k}.$$

Then it follows from (5.5) that

$$\left\| \frac{A^q}{\lambda^q} - \Lambda \right\|_\infty \le \frac{1}{\lambda^q} \|T\|_\infty \|T^{-1}\|_\infty \max_{i\ge 2} \sum_{k=0}^{s_i-1} \binom{q}{k} |\lambda_i|^{q-k},$$

and the corollary follows from Theorem 5.2. □

6. Stabilization for block encoders. In this section, we present an algorithm which computes $M_{\text{blk}}(q)$ and M^*_{blk} together with the achieving sets of principal states. Stabilization of block encoders is also studied.

Let G be a labeled graph. Define $T_G(w,v)$ to be the subset of states of G which are the terminal states of the paths labeled by w starting from state v. (Note that $T_G(w,v)$ has only one state if G is deterministic.)

Definition 6.1. *Let G be a labeled graph. We define $\bar{G}$ to be the graph with $V_{\bar{G}}$ being the set of all nonempty subsets of V_G, with an edge from U to V labeled by w if*

1. *for each $u \in U$, there is an outgoing edge with label w,*
2. $\bigcup_{u\in U} T_G(w,u) = V$.

We denote by $\bar{A}$ the adjacency matrix of $\bar{G}$.

This graph $\bar{G}$ is typically reducible and is closely related to the subset construction in finite automata theory. Note that $S(\bar{G}) = S(G)$. Moreover, $\bar{G}$ is always deterministic.

Example 6.2. The Shannon cover G and the corresponding $\bar{G}$ of RLL(1,2) are shown in Figures 6.1 and 6.2. By viewing each state u as a singleton subset $\{u\}$, we see that G is a subgraph of $\bar{G}$. (This is true for any deterministic graph.)

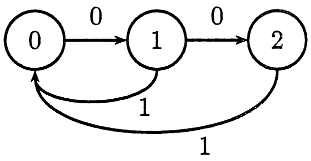


FIG. 6.1. *Shannon cover G of RLL*$(1,2)$.

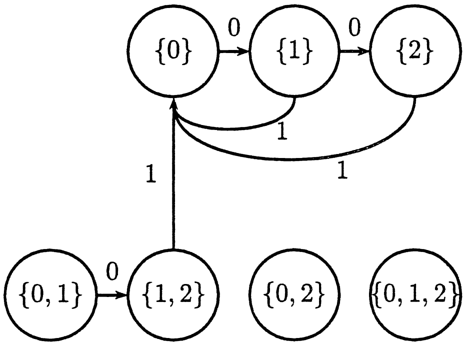


FIG. 6.2. $\bar{G}$ *for RLL*$(1,2)$.

LEMMA 6.3.

$$\overline{G^q} = (\bar{G})^q.$$

Proof. First observe that the sets of vertices of $\overline{G^q}$ and $(\bar{G})^q$ are the same. Next, we can see that there is an edge from U to V in $\overline{G^q}$ if and only if there is a sequence w and a path of length q labeled by w from every state $u \in U$ such that $\bigcup_{u\in U} T_G(w,u) = V$. The same is true for $(\bar{G})^q$. Because $\overline{G^q}$ and $(\bar{G})^q$ are deterministic by construction, the edge is unique and we can conclude that $\overline{G^q} = (\bar{G})^q$. □

The next theorem shows how to compute $M_{\text{blk}}(q,P)$ from $\bar{A}$.

THEOREM 6.4. *Let S be a constrained system and let G be a deterministic presentation of S. Let $\bar{A}$ be the adjacency matrix of $\bar{G}$. Then*

$$M_{\text{blk}}(q,P) = \sum_{U\subseteq P} \bar{A}^q_{P,U}.$$

Proof. From the definition of $M_{\text{blk}}(q,P)$, a word that can be counted for $M_{\text{blk}}(q,P)$ must be generated by an edge from every state in P and the terminal state for this edge must be in P. Hence,

$$\begin{aligned} M_{\text{blk}}(q,P) &= \left|\bigcup_{U\subseteq P} \mathcal{F}^1_{\overline{G^q}}(P,U)\right| = \left|\bigcup_{U\subseteq P} \mathcal{F}^1_{(\bar{G})^q}(P,U)\right| \qquad \text{(by Lemma 6.3)} \\ &= \left|\bigcup_{U\subseteq P} \mathcal{F}^q_{(\bar{G})}(P,U)\right| = \sum_{U\subseteq P} \bar{A}^q_{P,U}, \end{aligned}$$

where the last equality follows from the fact that $\bar{G}$ is deterministic. □

LEMMA 6.5. *Let S be a primitive constrained system and let G be the Shannon cover of S with adjacency matrix A. Then the adjacency matrix $\bar{A}$ of $\bar{G}$ has the following properties:*

(i) *$\bar{A}$ has a unique largest eigenvalue $\lambda = \lambda(A)$.*

(ii) *The right ($\bar{\mathbf{r}}$) and left ($\bar{\mathbf{l}}$) eigenvectors associated with λ are nonnegative. Furthermore, if the states of $\bar{G}$ are ordered so that the first $|V_G|$ states are of the form $\{u\}$, where $u \in V_G$ (subset of size one), then $\bar{\mathbf{r}}$ and $\bar{\mathbf{l}}$ have the form*

$$\bar{\mathbf{r}} = \begin{bmatrix} \mathbf{r} \\ \mathbf{s} \end{bmatrix}, \quad \bar{\mathbf{l}} = \begin{bmatrix} \mathbf{l} & 0 \end{bmatrix},$$

where $\mathbf{r}$ and $\mathbf{l}$ are the right and left eigenvectors of A associated with λ.

(iii) *Suppose $\bar{\mathbf{r}}$ and $\bar{\mathbf{l}}$ are normalized so that $\bar{\mathbf{l}}\bar{\mathbf{r}} = 1$ (equivalently $\mathbf{l}\mathbf{r} = 1$), and define $\bar{\Lambda} = \bar{\mathbf{r}}\bar{\mathbf{l}}$. Then $\lim_{q\to\infty} \frac{\bar{A}^q}{\lambda^q} = \bar{\Lambda}$.*

Proof.

(i) Because G is deterministic, G is a subgraph of $\bar{G}$. In particular, G is an irreducible component of $\bar{G}$. Since G is the Shannon cover of S, there must be a *homing word* h for a state in G [10, Lemma 2.10] (i.e., all paths in G that generate h must terminate in the same state). Let H be another irreducible component of $\bar{G}$. Then H cannot generate h because any path with label h must end in G. Therefore $S(H)$ is a proper subset of S. Thus $\lambda(A_H) < \lambda$ by [9, Corollary 4.4.9]. Since the set of eigenvalues of $\bar{A}$ is the union of the sets of eigenvalues of the adjacency matrices of the irreducible components of $\bar{G}$, we conclude that λ is the unique largest eigenvalue of $\bar{A}$.

(ii) It is easy to see that $\bar{A}$ has the form

$$\bar{A} = \begin{bmatrix} A & 0 \\ C & D \end{bmatrix}.$$

Let $\bar{\mathbf{l}} = \begin{bmatrix} \bar{\mathbf{l}}_1 & \bar{\mathbf{l}}_2 \end{bmatrix}$. Then the left eigenvector equation is

$$\begin{bmatrix} \bar{\mathbf{l}}_1 A + \bar{\mathbf{l}}_2 C & \bar{\mathbf{l}}_2 D \end{bmatrix} = \lambda \begin{bmatrix} \bar{\mathbf{l}}_1 & \bar{\mathbf{l}}_2 \end{bmatrix}.$$

From (i), λ is larger than all eigenvalues of D. Thus $\bar{\mathbf{l}}_2 = 0$. Moreover, $\bar{\mathbf{l}}_1 = \mathbf{l}$ is the left eigenvector of A corresponding to λ.

On the other hand, let

$$\bar{\mathbf{r}} = \begin{bmatrix} \bar{\mathbf{r}}_1 \\ \bar{\mathbf{r}}_2 \end{bmatrix}.$$

Then the right eigenvector equation is

$$\begin{bmatrix} A\bar{\mathbf{r}}_1 \\ C\bar{\mathbf{r}}_1 + D\bar{\mathbf{r}}_2 \end{bmatrix} = \begin{bmatrix} \lambda\bar{\mathbf{r}}_1 \\ \lambda\bar{\mathbf{r}}_2 \end{bmatrix}.$$

This implies that $\bar{\mathbf{r}}_1 = \mathbf{r}$ is the right eigenvector of A associated with λ and

$$\begin{aligned}
(\lambda I - D)\bar{\mathbf{r}}_2 &= C\mathbf{r}, \\
\bar{\mathbf{r}}_2 &= (\lambda I - D)^{-1} C\mathbf{r} \\
&= \lambda^{-1}\left(I + \frac{D}{\lambda} + \frac{D^2}{\lambda^2} + \cdots\right) C\mathbf{r} \\
&\geq 0.
\end{aligned}$$

(iii) $\bar{A}$ can be transformed into Jordan canonical form as

$$\bar{A} = \begin{bmatrix} \bar{\mathbf{r}} & R \end{bmatrix} \begin{bmatrix} \lambda & 0 \\ 0 & J \end{bmatrix} \begin{bmatrix} \bar{\mathbf{l}} \\ L \end{bmatrix},$$

where J comprises eigenvalues of D and A not equal to λ. From (i), all eigenvalues of D have magnitude less than λ. Moreover, it can be shown that the Shannon cover of a primitive constraint is primitive. Thus all eigenvalues of A not equal to λ have magnitude less than λ. Therefore

$$\bar{A}^q = \bar{\mathbf{r}}\bar{\mathbf{l}}\lambda^q + o(\lambda^q),$$

where $\lim_{q\to\infty} o(\lambda^q)/\lambda^q = 0$. Then the result follows. □

The following gives a characterization of M^*_{blk}.

THEOREM 6.6. *Let* $\bar{\mathbf{r}}$ *and* $\bar{\mathbf{l}}$ *be as in* (iii) *of Lemma* 6.5. *For a primitive constrained system,*

$$M^\infty_{\text{blk}}(P) = \bar{\mathbf{r}}_P \sum_{u\in P} \bar{\mathbf{l}}_{\{u\}}.$$

Moreover,

(i) $\mathcal{P}_{\text{blk}}(q) \subseteq \mathcal{P}^\infty_{\text{blk}}$ *for sufficiently large* q,

(ii) $M^*_{\text{blk}} = \max_{P\subseteq V_G} \left(\bar{\mathbf{r}}_P \sum_{u\in P} \bar{\mathbf{l}}_{\{u\}}\right)$.

Proof. From Theorem 6.4, $M^q_{\text{blk}}(P) = \frac{1}{\lambda^q}\sum_{U\subseteq P} \bar{A}^q_{P,U}$. Thus from (iii) of Lemma 6.5, we have

$$M^\infty_{\text{blk}}(P) = \lim_{q\to\infty} M^q_{\text{blk}}(P) = \sum_{U\subseteq P} \bar{\Lambda}_{P,U} = \bar{\mathbf{r}}_P \sum_{U\subseteq P} \bar{\mathbf{l}}_U.$$

Since $\bar{\mathbf{l}} = \begin{bmatrix} \mathbf{l} & 0 \end{bmatrix}$,

$$M^\infty_{\text{blk}}(P) = \bar{\mathbf{r}}_P \sum_{u\in P} \bar{\mathbf{l}}_{\{u\}}.$$

Then (i) and (ii) follow from Proposition 4.1. □

THEOREM 6.7. *If* q *satisfies*

$$\left\| \frac{\bar{A}^q}{\lambda^q} - \bar{\Lambda} \right\|_\infty < \frac{\epsilon_{\text{blk}}}{2},$$

then $\mathcal{P}_{\text{blk}}(q) \subseteq \mathcal{P}^\infty_{\text{blk}}$.

Proof. Let P be any set of principal states and let $\mathbf{x} = (x_U)$ be a 0-1 vector of dimension $|V_{\bar{G}}|$ such that $x_U = 1$ if $U \subseteq P$ and $x_U = 0$ otherwise. Then

$$\begin{aligned} |M^q_{\text{blk}}(P) - M^\infty_{\text{blk}}(P)| &= \left| \left(\frac{\bar{A}^q}{\lambda^q}\mathbf{x}\right)_P - (\bar{\Lambda}\mathbf{x})_P \right| \\ &\le \left\| \frac{\bar{A}^q}{\lambda^q} - \bar{\Lambda} \right\|_\infty < \frac{\epsilon_{\text{blk}}}{2}. \end{aligned}$$

Then the theorem follows from Lemma 4.2. □

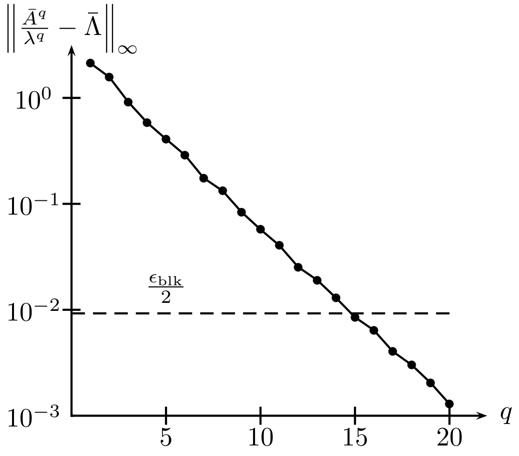


FIG. 6.3. $\|\frac{\bar{A}^q}{\lambda^q} - \bar{\Lambda}\|_\infty$.

Example 6.8. Consider the asymmetric-RLL$(2,5,1,3)$ described in Example 5.3. It is found that $M^*_{\text{blk}} = 0.3445$ and the only asymptotically optimal sets of principal states are $P^*_{\text{blk1}} = \{2, 3, \bar{1}\}$ and $P^*_{\text{blk2}} = \{2, \bar{1}, \bar{2}\}$. The second largest $M^\infty_{\text{blk}}(P)$ is 0.3260 when $P = \{2, \bar{1}\}$ and $\{2, 3, \bar{1}, \bar{2}\}$. Therefore $\epsilon_{\text{blk}} = 0.3445 - 0.3260 = 0.0185$.

We plot $\|\frac{\bar{A}^q}{\lambda^q} - \bar{\Lambda}\|_\infty$ in Figure 6.3. The plot suggests that $\|\frac{\bar{A}^q}{\lambda^q} - \bar{\Lambda}\|_\infty < \epsilon_{\text{blk}}/2$ for $q \geq 15$. This would imply that either P^*_{blk1} or P^*_{blk2} (or both) is an optimal set of principal states for $q \geq 15$.

The set of eigenvalues of $\bar{A}$ comprises the eight eigenvalues of A, all of which are nonzero and have multiplicity 1, and a zero eigenvalue with large multiplicity. Computing a transformation matrix for a matrix with an eigenvalue having such a large multiplicity is unstable; thus the idea in Corollary 5.4 cannot be directly applied. However, since the Shannon cover G has memory 5, all paths of length $q \geq 5$ in G that carry the same label must terminate at the same state. Therefore, assuming $q \geq 5$, every path of length q in $\bar{G}$ must terminate at a state of the form $\{u\}$ (a singleton state). Hence, $\bar{A}^q$ has only eight nonzero columns (that correspond to the singleton subsets). It follows that the Jordan blocks of $\bar{A}^q$ that correspond to the zero eigenvalue become zero. For this reason, when $q \geq 5$, we can write

$$\bar{A}^q = \bar{R}\bar{D}^q\bar{L} = \begin{bmatrix} R & 0 \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}^q \begin{bmatrix} L \\ 0 \end{bmatrix},$$

where D is the diagonal matrix containing the eigenvalues of A, and R and L contain the right and left eigenvectors of $\bar{A}$ corresponding to these eigenvalues, normalized so that LR is the identity matrix. Now we apply Theorem 6.7:

$$\begin{aligned}\left\|\frac{\bar{A}^q}{\lambda^q} - \bar{\Lambda}\right\|_\infty &\leq \|\bar{R}\|_\infty \|\bar{L}\|_\infty \left(\frac{|\lambda_2|}{\lambda}\right)^q \\ &= (5.9628)(2.7878)\left(\frac{1.1271}{1.6372}\right)^q.\end{aligned}$$

If $q \geq 21$, then

$$\left\|\frac{\bar{A}^q}{\lambda^q} - \bar{\Lambda}\right\|_\infty \leq (16.6233)(0.6884)^{21} = 0.0065 < 0.0092 = \frac{\epsilon_{\text{blk}}}{2}.$$

(We remark that $\|\bar{R}\|_\infty \|\bar{L}\|_\infty$ is not unique; different normalization of L and R gives different $\|\bar{R}\|_\infty \|\bar{L}\|_\infty$.)

By explicitly computing $M_{\rm blk}(q, P)$ for $1 \le q \le 30$, we find that $P^*_{\rm blk1}$ and $P^*_{\rm blk2}$ are optimal for all q in that range except $q = 5$. Moreover, they both are the only optimal sets of principal states when $8 \le q \le 30$.

We can further analyze the block codes $\mathcal{L}_1$ and $\mathcal{L}_2$ supported by $P^*_{\rm blk1}$ and $P^*_{\rm blk2}$. One can show that $\mathcal{L}_1$ comprises words with prefix 001, 10, or 110, and suffix 100, 1000, or 01. Similarly, $\mathcal{L}_2$ comprises words with prefix 001, 0001, or 10, and suffix 100, 01, or 011. Thus, a word $\mathbf{w} = w_1 w_2 \cdots w_q$ is in $\mathcal{L}_1$ if and only if its reversal $w_q w_{q-1} \cdots w_1$ is in $\mathcal{L}_2$. Therefore $M_{\rm blk}(q, P^*_{\rm blk1}) = |\mathcal{L}_1| = |\mathcal{L}_2| = M_{\rm blk}(q, P^*_{\rm blk2})$. We can now conclude that $P^*_{\rm blk1}$ and $P^*_{\rm blk2}$ are optimal for all q except $q = 5$ and are the only optimal sets of principal states when $q \ge 8$.

7. Stabilization for block-decodable encoders.

Among block-type-decodable encoders, we know the least about block-decodable encoders. In this section, we show that $M^\infty_{\rm blkdec}(P)$ and $M^*_{\rm blkdec}$ exist for any primitive constraint. Computation of asymptotically optimal sets of principal states is described. We also give a bound on q such that $\mathcal{P}_{\rm blkdec}(q) \subseteq \mathcal{P}^\infty_{\rm blkdec}$.

First consider the following input tag assignment problem. For a given deterministic graph G, we wish to find a block-decodable encoder that is a subgraph of G and has the same set of states as G. We can proceed as follows.

```
INPUT TAG ASSIGNMENT
    $\Psi \leftarrow$ set of all edge labels of $G$
    $\tau \leftarrow 1$
    while (it is possible to choose a set of edge labels $\psi = \{w_1, \ldots, w_k\} \subseteq \Psi$
            such that each state of $G$ can generate at least one $w_i$)
            do assign tag $\tau$ to each label in $\psi$
                $\tau \leftarrow \tau + 1$
                $\Psi \leftarrow \Psi \setminus \psi$
```

After the assignment, we obtain a desired encoder by keeping outgoing edges with distinct labels at each state and removing the other edges.

If we choose ψ wisely, the algorithm will give an optimal block-decodable encoder. Unfortunately, it is not clear how to choose ψ to maximize the number of tags; thus an algorithm to choose ψ is needed. We will use integer and linear programming to tackle this problem. Because the upcoming formulation of the integer programming problem involves many complex notations, we offer the following example to illustrate the idea.

Example 7.1. Let S be the constrained system presented by G in Figure 7.1. To simplify the figure, we draw only one edge for parallel edges. For example, state I has two edges to state J labeled by w_3 and w_4.

We wish to find an optimal block-decodable encoder for S. First we fix the set of principal states $P = \{I, J, K\}$ and compute $M_{\rm blkdec}(1, P)$. Consider the subgraph of G with the set of states P. We divide the labels of this subgraph into groups so that labels are in the same group if the sets of states that can generate them are equal. The diagram in Figure 7.2 summarizes this.

From the diagram, only I can generate w_3, only I and J can generate u, w_4, w_5, and so on. We will denote each region in the diagram by a subset of P; for example, the region that contains w_2 and w_6 is denoted by $\{I, K\}$.

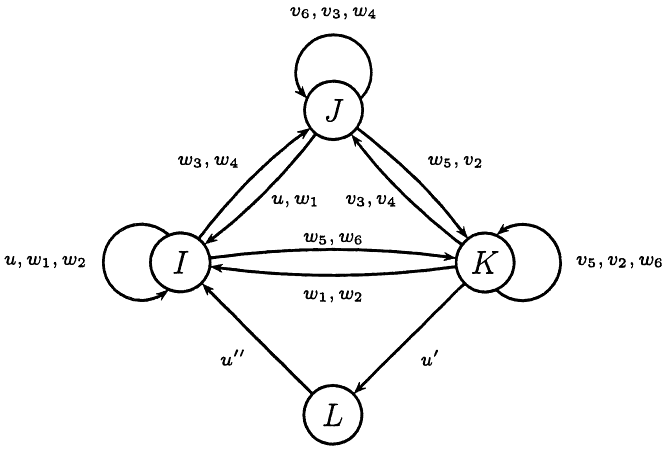


FIG. 7.1. *G in Example* 7.1.

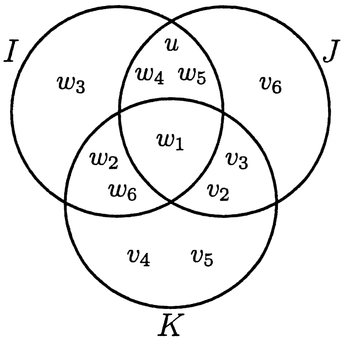


FIG. 7.2. *Partition of labels based on initial states for Example* 7.1.

From the input tag assignment algorithm, we choose a set of labels such that each state in P can generate at least one label. For instance, we can choose $\{w_1\}$ because every state in P can generate w_1. Then we assign tag 1 to all edges labeled by w_1. Also, we can choose $\{v_2, w_2\}$ because I and K can generate w_2 and J and K can generate v_2. So we assign tag 2 to all edges labeled by v_2 and w_2. Choosing a set of labels like this determines a cover of P. For example, choosing $\{w_1\}$ determines $\{\{I, J, K\}\}$. Also, choosing $\{v_2, w_2\}$ determines $\{\{J, K\}, \{I, K\}\}$. To obtain an optimal encoder, we only need to choose a set of labels that determines a minimal cover of P, that is, a cover for which removing a single member destroys the covering property [15].

For the design of codes, it can be seen that only the number of labels in each region is needed. For this reason, we further simplify the diagram to Figure 7.3.

It can be seen that there are eight minimal covers of P. We denote cover i by a 0-1 vector $\mathbf{z}_i = (z_U)$ of size $2^{|P|} - 1 = 7$ indexed by subsets of P such that $z_U = 1$ if U is in the cover and $z_U = 0$ otherwise. Let c_i denote the number of times that we choose cover i. Then the input tag assignment problem becomes an integer programming problem:

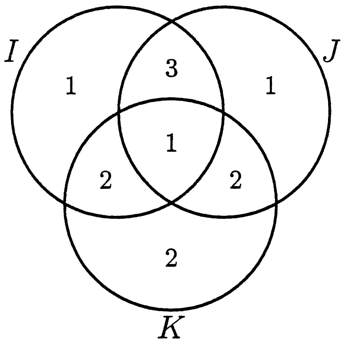


FIG. 7.3. *Number of labels based on initial states for Example* 7.1.

maximize $c_1 + c_2 + \cdots + c_8$
subject to $c_i \in \mathbb{Z}$,
$c_i \geq 0$,

$$\sum_{i=1}^{8} c_i \mathbf{z}_i = \begin{bmatrix} 1&1&0&0&0&0&0&0\\ 1&0&1&0&0&0&0&0\\ 1&0&0&1&0&0&0&0\\ 0&0&0&1&1&1&0&0\\ 0&0&1&0&1&0&1&0\\ 0&1&0&0&0&1&1&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix} \begin{bmatrix} c_1\\ c_2\\ c_3\\ c_4\\ c_5\\ c_6\\ c_7\\ c_8 \end{bmatrix} \leq \begin{bmatrix} 1\\ 1\\ 2\\ 3\\ 2\\ 2\\ 1 \end{bmatrix}. \tag{7.1}$$

A solution to this problem is $(c_i) = [\ 0\ \ 1\ \ 1\ \ 2\ \ 0\ \ 0\ \ 1\ \ 1\ ]^T$ which gives $M_{\text{blkdec}}(1, P) = \sum_i c_i = 6$. This solution can be achieved by assigning tag j to the edges labeled by v_j or w_j. Compare this to $M_{\text{det}}(1, P) = 7$ (minimum of the sums of each circle in Figure 7.3) and $M_{\text{blk}}(1, P) = 1$ (the size of the region $\{I, J, K\}$).

It can be shown that no other sets of principal states can give $M_{\text{blkdec}}(1, P) = 6$. This can be easily checked by computing $M_{\text{det}}(1, P)$. The maximum of $M_{\text{det}}(1, P)$ for $P \neq \{I, J, K\}$ is 5; thus $M_{\text{blkdec}}(1, P) \leq 5$ for $P \neq \{I, J, K\}$.

Now we give a formal description of how to relate the input tag assignment problem to an integer programming problem. Let G be a deterministic presentation of a constrained system S. Suppose w is a word and $P \subseteq V_G$; we define

$$D_G(w, P) = \{u \in P \ : \ w \in \mathcal{F}_G(u, P)\}.$$

We say that two words w_1 and w_2 are equivalent with respect to P if $D_G(w_1, P) = D_G(w_2, P)$. Clearly this is an equivalence relation. Therefore all words can be grouped into classes; each class is identified with $D_G(w, P)$, a subset of P, where w belongs to that class. In Example 7.1, this is the same as arranging words in Figure 7.2.

Define $\mathbf{d}_G(q, P)$ to be the $2^{|P|} - 1$ tuple indexed by nonempty subsets of P: for each $U \subseteq P$, $d_G(q, P)_U = |\{w \ : \ D_G(w, P) = U \text{ and } |w| = q\}|$, i.e., the number of words of length q in class U. In Example 7.1, this $\mathbf{d}_G(q, P)$ represents the vector in the right-hand side of (7.1).

We claim that $\mathbf{d}_G(q, P)$ is determined by $\bar{A}^q$. To see this, let M be a $(2^{|V_G|} - 1) \times (2^{|V_G|} - 1)$ matrix indexed by the nonempty subsets of V_G. For each nonempty

$U \subseteq P$, let $\gamma_M(U,P) = \sum_{V \subseteq P} M_{U,V}$. In particular when $M = \bar{A}^q$, $\gamma_{\bar{A}^q}(U,P)$ is the number of words of length q that can be generated by every state in U with terminal state in P. Note that $\gamma_{\bar{A}^q}(U,P)$ overcounts $d_G(q,P)_U$ because it also counts words generated from proper supersets of U. To compute $\mathbf{d}_G(q,P)$, define

$$\Delta(M,U,P) = \gamma_M(U,P) - \sum_{\{v\} \subseteq P \setminus U} \gamma_M(U \cup \{v\}, P)$$

$$+ \sum_{\{v_1,v_2\} \subseteq P \setminus U} \gamma_M(U \cup \{v_1,v_2\}, P) - \cdots (-1)^{|P|-|U|} \gamma_M(P,P).$$

Then it follows from the principle of inclusion and exclusion that

$$d_G(q,P)_U = \Delta(\bar{A}^q, U, P).$$

Define $\mathbf{d}_G^\infty(P) = \lim_{q \to \infty} \mathbf{d}_G(q,P)/\lambda^q$. It follows from (iii) of Lemma 6.5 that if S is primitive and G is the Shannon cover of S, then $\mathbf{d}_G^\infty(P)$ exists and $d_G^\infty(P)_U = \Delta(\bar{\Lambda}, U, P)$.

By following the idea in Example 7.1, we view the classes of words as subsets of P. Then we choose a minimal cover of P which can be represented by the vector $\mathbf{z}$. Let $t = t(|P|)$ be the number of minimal covers of P. ($t = 8$ in Example 7.1.) Then the problem of finding an input tag assignment which achieves $M_{\text{blkdec}}(q,P)$ becomes an integer programming problem:

maximize	$c_1 + c_2 + \cdots + c_t$,
subject to	$c_i \in \mathbb{Z}$,
	$c_i \geq 0$,
	$c_1\mathbf{z}_1 + c_2\mathbf{z}_2 + \cdots + c_t\mathbf{z}_t \leq \mathbf{d}_G(q,P)$.

If we delete the first condition, this becomes a linear programming problem. We view the maximum of the objective function of this relaxed problem as a function $\mu(\mathbf{x})$ whose argument $\mathbf{x}$ represents $\mathbf{d}_G(q,P)$ above. ($\mathbf{x}$ is allowed to be real.) So the value of $\mu(\mathbf{x})$ is $\sum_{i=1}^t c_i$, where (c_i) is a solution to the relaxed problem. This defines $\mu(\mathbf{x})$ for a vector $\mathbf{x}$ of fixed dimension. We can generalize the domain of μ to include all nonnegative real vectors with dimension of the form $2^n - 1$, $1 \leq n \leq |V_G|$. In this way, we define $\mu(\mathbf{d}_G(q,P))$ for any P. We can show the following properties of μ.

PROPOSITION 7.2. *Let $\mathbb{R}_{\geq 0}$ denote the set of nonnegative reals.*

(i) *$\mu(a\mathbf{x}) = a\mu(\mathbf{x})$ for any $a \in \mathbb{R}_{\geq 0}$.*

(ii) *$|\mu(\mathbf{x}) - \mu(\mathbf{y})| \leq \|\mathbf{x} - \mathbf{y}\|_1$ for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}_{\geq 0}^{2^n-1}$.*

Proof.

(i) Since the case $a = 0$ is trivially true, we assume that $a > 0$. Suppose $\mathbf{c} = (c_i)$ is a solution to the linear programming problem with input $\mathbf{x}$. Then $a\mathbf{c}$ satisfies the condition of the problem when the input is $a\mathbf{x}$. Thus $\mu(a\mathbf{x}) \geq a \sum_i c_i = a\mu(\mathbf{x})$. Using the same argument with $\mathbf{x}$ replaced by $a\mathbf{x}$ and a replaced by $1/a$, we can show that $\mu(a\mathbf{x}) \leq a\mu(\mathbf{x})$. Therefore $\mu(a\mathbf{x}) = a\mu(\mathbf{x})$.

(ii) It is sufficient to show this when $\mathbf{x}$ and $\mathbf{y}$ differ at only one entry; then the proposition follows from the triangle inequality. Suppose $\mathbf{x}$ and $\mathbf{y}$ differ at only the jth entry. Without loss of generality, assume $x_j > y_j$. Suppose $\mathbf{c}$ is a solution to the problem when $\mathbf{x}$ is the input. Consider the sum $\sum_{i=1}^t c_i(z_i)_j$, which must be less than or equal to x_j. If it is less than or equal to y_j, then $\mathbf{c}$ is also a solution when $\mathbf{y}$ is the input and we have $\mu(\mathbf{y}) = \mu(\mathbf{x})$. If the sum is greater than y_j, we find a vector $\mathbf{c}'$ as follows. Let $I = \{i : (z_i)_j = 1\}$. Let $\mathbf{c}'$ be a vector such that $c'_i \geq 0$ for all $1 \leq i \leq t$, $c'_i = c_i$ for $i \notin I$, and $\sum_{i \in I} c'_i = y_j$. The vector $\mathbf{c}'$ satisfies the condition of the problem when $\mathbf{y}$ is the input. Therefore $\mu(\mathbf{x}) - \mu(\mathbf{y}) \leq \sum_{i=1}^t c_i - \sum_{i=1}^t c'_i = \sum_{i \in I} c_i - \sum_{i \in I} c'_i \leq x_j - y_j$, and the proposition is proved. □

We note that from (ii) above, μ is uniformly continuous.

PROPOSITION 7.3.

$$M_{\mathrm{blkdec}}(q, P) \leq \mu(\mathbf{d}_G(q, P)) \leq M_{\mathrm{blkdec}}(q, P) + t.$$

Proof. Recall that $M_{\mathrm{blkdec}}(q, P)$ is the maximum of the objective function in the integer programming problem above. Since the domain of the variables is more restricted (to integers rather than reals), $M_{\mathrm{blkdec}}(q, P) \leq \mu(\mathbf{d}_G(q, P))$.

Suppose that a vector $\mathbf{c} = (c_i)$ is a solution to the linear programming problem. Then $\mu(\mathbf{d}_G(q, P)) = \sum_{i=1}^t c_i$, and $\lfloor \mathbf{c} \rfloor = (\lfloor c_i \rfloor)$ satisfies the condition in the integer programming problem. Thus

$$M_{\mathrm{blkdec}}(q, P) \geq \sum_{i=1}^t \lfloor c_i \rfloor \geq \sum_{i=1}^t c_i - t = \mu(\mathbf{d}_G(q, P)) - t. \qquad \square$$

THEOREM 7.4. *For a primitive constrained system,*

$$M^\infty_{\mathrm{blkdec}}(P) = \mu(\mathbf{d}^\infty_G(P)).$$

Moreover,

(i) $\mathcal{P}_{\mathrm{blkdec}}(q) \subseteq \mathcal{P}^\infty_{\mathrm{blkdec}}$ *for sufficiently large* q,

(ii) $M^*_{\mathrm{blkdec}} = \max_{P \subseteq V_G} \mu(\mathbf{d}^\infty_G(P))$.

Proof.

$$\lim_{q \to \infty} \frac{\mu(\mathbf{d}_G(q, P))}{\lambda^q} = \lim_{q \to \infty} \mu\left(\frac{\mathbf{d}_G(q, P)}{\lambda^q}\right) \qquad \text{(by (i) of Proposition 7.2)}$$

$$= \mu(\mathbf{d}^\infty_G(P)) \qquad \text{(since } \mu \text{ is continuous)}.$$

From Proposition 7.3, $M_{\mathrm{blkdec}}(q, P)/\lambda^q$ also converges to the same limit. Then (i) and (ii) follow from Proposition 4.1. □

Next we give a bound on q similar to Theorems 5.2 and 6.7. Recall that $t(k)$ is the number of minimal covers of a set of size k. Define

$$\rho(G, q) = (2^{|V_G|} - 1) \sum_{U,V} \left| \left(\frac{\bar{A}^q}{\lambda^q}\right)_{U,V} - \bar{\Lambda}_{U,V} \right| + \frac{t(|V_G|)}{\lambda^q}.$$

Note that $\lim_{q \to \infty} \rho(G, q) = 0$ because $\frac{\bar{A}^q}{\lambda^q}$ converges to $\bar{\Lambda}$.

THEOREM 7.5. *If q satisfies $\rho(G,q) < \epsilon_{\text{blkdec}}/2$, then $\mathcal{P}_{\text{blkdec}}(q) \subseteq \mathcal{P}^{\infty}_{\text{blkdec}}$.*

Proof. First observe that $\Delta(\bar{A}^q, W, P)$ can be written as $\sum_{U,V} a_{U,V} \bar{A}^q_{U,V}$, where $a_{U,V} \in \{0, 1, -1\}$. Thus for any W,

$$\begin{aligned}\frac{d_G(q,P)_W}{\lambda^q} - d_G^{\infty}(P)_W &= \sum_{U,V} a_{U,V} \left(\left(\frac{\bar{A}^q}{\lambda^q} \right)_{U,V} - \bar{\Lambda}_{U,V} \right) \\ &\leq \sum_{U,V} \left| \left(\frac{\bar{A}^q}{\lambda^q} \right)_{U,V} - \bar{\Lambda}_{U,V} \right| .\end{aligned}$$

Therefore

$$\begin{aligned}&|M^q_{\text{blkdec}}(P) - M^{\infty}_{\text{blkdec}}(P)| \\ &\leq \left| \frac{\mu(\mathbf{d}_G(q,P))}{\lambda^q} - \mu(\mathbf{d}_G^{\infty}(P)) \right| + \left| \frac{M_{\text{blkdec}}(q,P)}{\lambda^q} - \frac{\mu(\mathbf{d}_G(q,P))}{\lambda^q} \right| \\ &\leq \left| \mu\left(\frac{\mathbf{d}_G(q,P)}{\lambda^q} \right) - \mu(\mathbf{d}_G^{\infty}(P)) \right| + \frac{t(|P|)}{\lambda^q} \\ &\leq \left\| \frac{\mathbf{d}_G(q,P)}{\lambda^q} - \mathbf{d}_G^{\infty}(P) \right\|_1 + \frac{t(|P|)}{\lambda^q} \\ &\leq (2^{|P|} - 1) \sum_{U,V} \left| \left(\frac{\bar{A}^q}{\lambda^q} \right)_{U,V} - \bar{\Lambda}_{U,V} \right| + \frac{t(|P|)}{\lambda^q} \\ &\leq (2^{|V_G|} - 1) \sum_{U,V} \left| \left(\frac{\bar{A}^q}{\lambda^q} \right)_{U,V} - \bar{\Lambda}_{U,V} \right| + \frac{t(|V_G|)}{\lambda^q} \\ &= \rho(G,q) < \frac{\epsilon_{\text{blkdec}}}{2}.\end{aligned}$$

Then the theorem follows from Lemma 4.2. □

With the technique described in this section, we can compute upper and lower bounds for $M_{\text{blkdec}}(q,P)$. The upper bound comes from the relaxed linear programming problem. The lower bound is obtained by "rounding down" the solution of the linear programming problem. Thus by checking all sets of principal states, we can obtain upper and lower bounds for $M_{\text{blkdec}}(q)$. In fact, from Proposition 3.1, it is sufficient to check all complete sets. Given a deterministic graph G and an integer n, Marcus, Siegel, and Wolf [11] gave an algorithm to find all complete sets P such that $M_{\text{det}}(1,P) \geq n$. Therefore we can design a block-decodable encoder as follows.

Given a deterministic presentation G of the desired constraint S and a block length q, find an optimal set of principal states P_{det} for a deterministic encoder. Then compute the upper and lower bounds for $M_{\text{blkdec}}(q, P_{\text{det}})$; set n to be the lower bound. For each complete set P such that $M_{\text{det}}(q,P) \geq n$, compute the upper and lower bounds for $M_{\text{blkdec}}(q,P)$. Set the upper and lower bounds for $M_{\text{blkdec}}(q)$ to be the maximum of the upper bounds and the maximum of the lower bounds for $M_{\text{blkdec}}(q,P)$, respectively. In this way, we also have the candidates for the optimal sets of principal states.

Example 7.6. By solving the linear programming problem described in this section, we find the asymptotically optimal set of principal states for the asymmetric-RLL$(2,5,1,3)$ to be $P^*_{\text{blkdec}} = \{1,2,3,\bar{1},\bar{2}\}$ and $M^*_{\text{blkdec}} = 0.7076$. Moreover, $\epsilon_{\text{blkdec}} = 0.0146$.

Next we apply Theorem 7.5 to compute the bound on q such that P^*_{blkdec} achieves $M_{\text{blkdec}}(q)$. Let $\bar{\mathbf{r}}_i$ and $\bar{\mathbf{l}}_i$, $1 \le i \le 8$, be the right and left eigenvectors corresponding to λ_i. From [14, Sequence A046165], $t(8) = 3731508$. Thus

$$\begin{aligned}\rho(G,q) &= (2^{|V_G|}-1)\sum_{U,V}\left|\left(\frac{\bar{A}^q}{\lambda^q}\right)_{U,V} - \bar{\Lambda}_{U,V}\right| + \frac{t(|V_G|)}{\lambda^q}\\ &= (255)\frac{1}{\lambda^q}\sum_{U,V}\left|\sum_{i=2}^{8}\lambda_i^q(\bar{r}_i\bar{l}_i)_{U,V}\right| + \frac{3731508}{\lambda^q}\\ &\le (255)\frac{1}{\lambda^q}\sum_{U,V}\sum_{i=2}^{8}|\lambda_i|^q|(\bar{r}_i\bar{l}_i)_{U,V}| + \frac{3731508}{\lambda^q}.\end{aligned}$$

This expression is decreasing with q. If $q \ge 44$, then $\rho(G,q) \le 0.0050 < \epsilon_{\text{blkdec}}/2 = 0.0073$.

In our construction of the integer programming problem described in this section, the number of variables depends only on $|P|$. But $\mathbf{d}_G(q,P)$ usually contains many zeros and thus many minimal covers of P are not necessary. Thus we can formulate an equivalent problem with a much smaller number of variables, and so the bound on q given in Theorem 7.5 can be very weak. This is especially true when the constraint has a lot of structure (e.g., when many follower sets can be ordered by inclusion). For this example, after neglecting the unnecessary minimal covers, the maximum number of variables is 14 when $P = \{1,2,3,4,\bar{1},\bar{2}\}$ while $t(8) = 3731508$.

Finally we compute bounds on $M_{\text{blkdec}}(q)$, $1 \le q \le 43$, as well as candidates for the achieving P. We find that P^*_{blkdec} is the only optimal set of principal states for $12 \le q \le 43$. From this computation and the bound on q explained above, we conclude that P^*_{blkdec} is the only optimal set of principal states for $q \ge 12$.

8. Complexity of block-type-decodability. We have studied some algorithms to design block-type-decodable encoders, and the reader may have noticed that finding an optimal deterministic encoder is easier than finding an optimal block encoder, which in turn is easier than finding an optimal block-decodable encoder. In this section, we study the complexity of these problems and show that, in some aspects, this observation is indeed the case.

Let S be a constrained system with a deterministic presentation G and let n be a positive integer. For each class $\mathcal{C}$ of encoders, we consider three slightly different problems.

1. *Subgraph encoder:* We study the complexity of determining whether there exists an (S,n) encoder in class $\mathcal{C}$ which is a subgraph of G. In this case, we aim to answer whether $M_{\mathcal{C}}(1) \ge n$. This is the most general and possibly the most important problem.

2. *Fully supported subgraph encoder:* We consider the same problem but require that the set of principal states be V_G. This case can be viewed as a subproblem of the first problem: we fix a set of principal states P and wish to determine whether $M_{\mathcal{C}}(1,P) \ge n$. We will see that this case distinguishes the complexity of computing block and block-decodable encoders.

3. $|V_G|$ *fixed:* In a practical encoder design, we usually fix the constraint and let the block length q vary; thus we study the first problem but consider the number of states of G to be fixed.

We remark that our goal is to compute $M_{\mathcal{C}}(1)$, but the complexity of this problem is equivalent to the complexity of determining whether $M_{\mathcal{C}}(1) \geq n$. We consider the latter problem because it is a decision problem, and hence its complexity class is easier to determine.

We begin with the problem of determining whether there exists an (S, n) deterministic encoder which is a subgraph of G. This problem can be solved by applying the Franaszek algorithm [4] to the adjacency matrix A of G and the all-ones vector $\mathbf{x}$. The algorithm proceeds by iteratively computing $\mathbf{x} \leftarrow \min \left\{ \left\lfloor \frac{1}{n} A\mathbf{x} \right\rfloor, \mathbf{x} \right\}$ (taken componentwise) until it converges. If $M_{\det}(1) < n$, the algorithm returns a zero vector; if $M_{\det}(1) \geq n$, the algorithm returns the characteristic vector of the largest set of principal states P such that $M_{\det}(1, P) \geq n$. It is easy to see that the algorithm terminates in at most $|V_G|$ iterations; in each iteration, the running time is polynomial. Thus, for the class of deterministic encoders, this problem is solvable in polynomial time.

From this result, it follows that the other two easier problems on deterministic encoders are also solvable in polynomial time. For the fully supported subgraph encoder problem, there exists an (S, n) deterministic encoder with the set of principal states V_G if and only if the Franaszek algorithm terminates in one iteration and returns the all-ones vector.

Next, we consider the class of block encoders. We will show that the problem of determining whether there exists an (S, n) block encoder is NP-complete by relating it to the well-known clique problem. We first describe the clique problem. A *k-clique* in an undirected graph is a subgraph with k nodes such that there is an edge between every two nodes in the clique. The clique problem is to determine whether a graph contains a clique of a specified size. This problem is known to be NP-complete [13].

Theorem 8.1. *Given a labeled graph G and an integer n, the problem of determining whether there exists an $(S(G), n)$ block encoder which is a subgraph of G is NP-complete. However, the problem becomes polynomial for every fixed n.*

Proof. Given a graph G' with output labeling and input tagging, it can be verified in polynomial time whether (i) G' is a subgraph of G and (ii) G' is an (S, n) block encoder. Therefore this problem is in NP. What remains is to show that the clique problem is polynomial-time reducible to this problem. Given an undirected graph $H = (V_H, E_H)$, we construct a labeled directed graph G as follows. Let $V_G = V_H$ and assign an edge from state u to state v labeled by v if H has an edge between u and v. Moreover, assign a self-loop to every state v labeled by v. One can show that H has an n-clique if and only if there exists an (S, n) block encoder which is a subgraph of G. Hence we conclude that the block encoder problem is NP-complete.

Suppose that n is fixed; we will show that the problem becomes polynomial. We choose n words from the set of all words and determine whether they can be concatenated with each other. If so, we have an (S, n) block encoder. If not, choose another set of n words. Since there are polynomially many ways to choose n words, we conclude that the problem is polynomial. ◻

If we require that the set of principal states of our block encoder be V_G, this problem becomes polynomial. To see this, consider the following polynomial-time algorithm.

```
COMPUTATION OF $M_{\text{blk}}(1, V_G)$
    **Input:** $G$ with $V_G = \{v_1, \ldots, v_m\}$
    $\mathcal{L} \leftarrow \mathcal{F}_G^1(v_1)$
    **for** each $2 \leq i \leq m$
        **for** each $w \in \mathcal{L}$
            **if** $w \notin \mathcal{F}_G^1(v_i)$
                **then** $\mathcal{L} \leftarrow \mathcal{L} \setminus \{w\}$
    **Output:** $|\mathcal{L}|$
```

For the third case where the number of states of G is fixed, the block encoder problem becomes polynomial because we can adapt the above algorithm for each set of principal states, and there is a fixed number of sets of principal states, namely, $2^{|V_G|} - 1$.

Finally, we turn to the block-decodability problem. The complexity of this problem has been studied by Ashley, Karabed, and Siegel [1]; the following theorem is a special case of [1, Theorem 5.4].

THEOREM 8.2 (see [1]). *Given a labeled graph G and an integer n, the problem of determining whether there exists an $(S(G), n)$ block-decodable encoder $\mathcal{E}$ which is a subgraph of G and $V_{\mathcal{E}} = V_G$ is NP-complete. It is also NP-complete for fixed $n \geq 2$.*

Thus the fully supported subgraph encoder problem for the block-decodable encoder is NP-complete. We will show that the subgraph encoder problem is also NP-complete by relating it to the fully supported subgraph encoder problem.

THEOREM 8.3. *Given a labeled graph G and an integer n, the problem of determining whether there exists an $(S(G), n)$ block-decodable encoder which is a subgraph of G is NP-complete. It is also NP-complete for fixed $n \geq 2$.*

Proof. This problem is easily seen to be in NP. We will show that the fully supported subgraph encoder problem is polynomial-time reducible to this more general problem. Given a graph H with $V_H = \{v_1, \ldots, v_m\}$, we construct another graph G as follows. Let $V_G = V_H$, and for each outgoing edge from v_i in H, we assign an edge in G from v_i to v_{i+1} with the same label. (If $i = m$, we assign an edge from v_m to v_1.) Clearly, if there is an $(S(H), n)$ block-decodable encoder $\mathcal{E}$ which is a subgraph of H and $V_{\mathcal{E}} = V_H$, then there is an $(S(G), n)$ block-decodable encoder which is a subgraph of G. On the other hand, if there is an $(S(G), n)$ block-decodable encoder which is a subgraph of G, this encoder must have the same set of states as G. By using the corresponding edges in H and the same tag assignment, we obtain an $(S(H), n)$ block-decodable encoder $\mathcal{E}$ which is a subgraph of H and $V_{\mathcal{E}} = V_H$. □

From Theorems 8.2 and 8.3, the block-decodability problem is generally intractable. However, if we fix the number of states but let only the number of edges and the size of the alphabet grow, then the problem becomes more tractable.

THEOREM 8.4. *Given a constrained system S with a deterministic presentation G and an integer n, the problem of determining whether there exists an (S, n) block-decodable encoder is solvable in polynomial time if we fix the number of states of G.*

Proof. First we fix a set of principal states P. It is sufficient to show that the problem of determining whether $M_{\text{blkdec}}(1, P) \geq n$ is solvable in polynomial time. This is because the number of sets of principal states is fixed $(= 2^{|V_G|} - 1)$.

In section 7, we showed that the computation of $M_{\text{blkdec}}(1, P)$ is equivalent to an integer programming problem. The worst-case number of variables of this problem (the largest t) depends only on $|P|$. Hence, we can consider the number of variables to be fixed.

Case 1. $n > |E_G|$. Clearly, we can conclude that $M_{\text{blkdec}}(1, P) < n$.

Case 2. $n \le |E_G|$. If we check the feasibility of all $\mathbf{c}$ such that $0 \le c_i \le n$, we can determine whether $M_{\text{blkdec}}(1, P) \ge n$. Since there are $(n+1)^t$ such $\mathbf{c}$, it can be checked in polynomial time. This is because $(n+1)^t \le (|E_G|+1)^t$ and t is fixed. $\square$

The complexity of each problem for each class of encoder is summarized in the following table.

Table 8.1
Complexity of block-type-decodability problems.

Encoder class	Subgraph encoder	Fully supported subgraph encoder	$\lvert V_G\rvert$ fixed
Deterministic	polynomial	polynomial	polynomial
Block	NP-complete (polynomial for any fixed n)	polynomial	polynomial
Block-decodable	NP-complete for fixed $n \ge 2$	NP-complete for fixed $n \ge 2$	polynomial

Acknowledgment. The authors are grateful to the anonymous referees for various comments and suggestions, particularly on the subject of minimal covers of finite sets which greatly improves section 7.

REFERENCES

[1] J. J. Ashley, R. Karabed, and P. H. Siegel, *Complexity and sliding-block decodability*, IEEE Trans. Inform. Theory, 42 (1996), pp. 1925–1947.

[2] J. Ashley and B. Marcus, *Canonical encoders for sliding block decoders*, SIAM J. Discrete Math., 8 (1995), pp. 555–605.

[3] P. Chaichanavong and B. H. Marcus, *Optimal block-type-decodable encoders for constrained systems*, IEEE Trans. Inform. Theory, 49 (2003), pp. 1231–1250.

[4] P. A. Franaszek, *Sequence-state coding for digital transmission*, Bell System Tech. J., 47 (1968), pp. 143–155.

[5] C. V. Freiman and A. D. Wyner, *Optimum block codes for noiseless input restricted channels*, Information and Control, 7 (1964), pp. 398–415.

[6] J. Gu and T. E. Fuja, *A new approach to constructing optimal block codes for runlength-limited channels*, IEEE Trans. Inform. Theory, 40 (1994), pp. 774–785.

[7] K. A. S. Immink, *Codes for Mass Data Storage Systems*, Shannon Foundation Publishers, Eindhoven, The Netherlands, 1999.

[8] P. Lee and J. K. Wolf, *A general error-correcting code construction for runlength limited binary channels*, IEEE Trans. Inform. Theory, 35 (1989), pp. 1330–1335.

[9] D. Lind and B. Marcus, *An Introduction to Symbolic Dynamics and Coding*, Cambridge University Press, Cambridge, UK, 1995.

[10] B. H. Marcus, R. M. Roth, and P. H. Siegel, *Constrained systems and coding for recording channels*, in Handbook of Coding Theory, Vols. I and II, North–Holland, Amsterdam, 1998, pp. 1635–1764.

[11] B. H. Marcus, P. H. Siegel, and J. K. Wolf, *Finite-state modulation codes for data storage*, IEEE J. Select. Areas Commun., 10 (1992), pp. 5–37.

[12] E. Seneta, *Nonnegative Matrices and Markov Chains*, 2nd ed., Springer Ser. Statist., Springer-Verlag, New York, 1981.

[13] M. Sipser, *Introduction to the Theory of Computation*, PWS, Boston, 1997.

[14] N. J. A. Sloane, *The On-Line Encyclopedia of Integer Sequences*, http://www.research.att.com/~njas/sequences (22 April 2005).

[15] E. W. Weisstein, *Minimal Cover*, from MathWorld—A Wolfram Web Resource, http://mathworld.wolfram.com/MinimalCover.html (2005).

SIAM J. DISCRETE MATH.
Vol. 19, No. 2, pp. 345–362

OPTIMAL AUGMENTATION FOR BIPARTITE COMPONENTWISE BICONNECTIVITY IN LINEAR TIME[*]

TSAN-SHENG HSU[†] AND MING-YANG KAO[‡]

Abstract. A graph is *componentwise biconnected* if every connected component either is an isolated vertex or is biconnected. We present a linear-time algorithm for the problem of adding the smallest number of edges to make a bipartite graph componentwise biconnected while preserving its bipartiteness. This algorithm has immediate applications for protecting sensitive information in statistical tables.

Key words. biconnectivity, data security, bipartite graph augmentation

AMS subject classifications. 68Q20, 68R10, 94C15, 05C40, 05C90

DOI. 10.1137/S0895480196303216

1. Introduction.

There is a long history of applications for the problem of adding edges to a graph in order to satisfy connectivity specifications (see [7, 10, 21] for recent examples). Correspondingly, the problem has been extensively studied for making general graphs k-edge connected or k-vertex connected for various values of k [5, 12, 11, 13, 14, 20, 29, 33] as well as for making vertex subsets suitably connected [6, 16, 18, 30, 31, 32, 34].

In this paper, we focus on augmenting bipartite graphs. A graph is *componentwise biconnected* if every connected component either is biconnected or is an isolated vertex. This paper presents a linear-time algorithm for the problem of inserting the smallest number of edges into a given bipartite graph to make it componentwise biconnected while maintaining its bipartiteness. This problem and related bipartite augmentation problems arise naturally from research on statistical data security [1, 2, 3, 4, 25]. To protect sensitive information in a cross tabulated table, it is a common practice to suppress some of the cells in the table. A basic issue concerning the effectiveness of this practice is how a table maker can suppress a small number of cells in addition to the sensitive ones so that the resulting table does not leak significant information. This protection problem can be reduced to augmentation problems for bipartite graphs [8, 17, 22, 23, 24, 26, 27, 28]. In particular, a linear-time algorithm for our augmentation problem immediately yields a linear-time algorithm for suppressing the smallest number of additional cells so that no nontrivial information about any individual row or column is revealed to an adversary [23].

Figure 1 gives an example to illustrate the relationship between our augmentation problem and the table protection problem. On the left is a 2-dimensional cross tabulated table with some suppressed cells. On the right is a bipartite *suppressed graph* constructed from the table, where the vertices correspond to the columns and rows,

[*] Received by the editors May 8, 1996; accepted for publication (in revised form) December 8, 2004; published electronically October 20, 2005. An extended abstract of this work appeared in *Proceedings of the 7th Annual International Symposium on Algorithms and Computation*, Springer-Verlag, New York, 1996, pp. 213–222.

http://www.siam.org/journals/sidma/19-2/30321.html

[†] Institute of Information Science, Academia Sinica, Nankang 11529, Taipei, Taiwan, Republic of China (tshsu@iis.sinica.edu.tw). This author's research was supported in part by NSC grants 85-2213-E-001-003, 86-2213-E-001-012, and 87-2213-E-001-022.

[‡] Department of Computer Science, Northwestern University, Evanston, IL 60201 (kao@cs.northwestern.edu). This author's research was supported in part by NSF grant CCR-9531028.

index	a	b	c	d	sum
1					21
2	4	3			18
3	7	3			18
4			1	6	17
sum	15	20	11	28	74

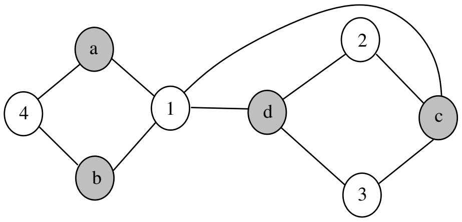


Fig. 1. *An example of the relationship between graph augmentation and table protection. On the left is a 2-dimensional cross tabulated table with some suppressed cells. On the right is the bipartite suppressed graph constructed from the table, where the vertices correspond to the columns and rows, and the edges correspond to the suppressed cells.*

and the edges correspond to the suppressed cells. Note that in the suppressed graph, row vertex 1 is the only cut vertex. It is proven in [23] that the value of any linear combination of the cells in a row or column that does not correspond to a cut vertex cannot be uniquely determined, except for the multiples of the sum of all suppressed cells in that row or column. Conversely, since vertex 1 is a cut vertex, the values of some linear combinations that are not multiples of the sum of the suppressed cells in row 1 can still be inferred from the information available in the table. For instance, let $C_{i,j}$ be the cell at the intersection of row i and column j, let $S_{*,j}$ be the sum of the cells in column j, and let $S_{i,*}$ be the sum of the cells in row i. Then, the value of $C_{1,a}+C_{1,b}$ must be 8 because it equals $S_{*,a}+S_{*,b}-\sum_{i=2}^{4}(C_{i,a}+C_{i,b})$, the value of $\sum_{i=2}^{3}(C_{i,a}+C_{i,b})$ is directly available from the table, and $C_{4,a}+C_{4,b}=S_{*,4}-C_{4,c}-C_{4,d}$.

Section 2 formally states our augmentation problem and discusses some main results. Section 3 proves an optimal bound on the smallest number of additional edges needed for the problem. Section 4 gives a linear-time algorithm to solve the augmentation problem.

2. Problem formulation, main results, and basic concepts. In this paper, all graphs are undirected and have neither self loops nor multiple edges.

2.1. The augmentation problem. Two vertices of a graph are *biconnected* if they are in the same connected component and remain so after the removal of any single edge or any single vertex other than either of them. A set of vertices is *biconnected* if every pair of its vertices are biconnected; similarly, a graph is *biconnected* if its set of vertices is biconnected. To suit our application of protecting sensitive information in statistical tables, this definition for biconnectivity is slightly different from the one used in standard textbooks. In particular, we define a connected component of an isolated vertex to be biconnected and one with exactly two vertices to be not biconnected.

A *block* of a graph is the induced subgraph of a maximal subset of vertices that is biconnected. A graph is *componentwise biconnected* if every connected component is a block. Throughout this paper, $G=(A,B,E)$ denotes a bipartite graph. A *legal edge* of G is an edge in $A\times B$ but not in E. A *biconnector* of G is a set L of legal edges such that $(A,B,E\cup L)$ is componentwise biconnected. An *optimal* biconnector is one with the smallest number of edges. Note that if $A=\emptyset$ or $B=\emptyset$, G is componentwise biconnected. If $|A|=1$ and $B\neq\emptyset$ (or $|B|=1$ and $A\neq\emptyset$), G has no biconnector. If $|A|\geq 2$ and $|B|\geq 2$, G has a biconnector. In light of these observations, the *optimal biconnector problem* is the following: given $G=(A,B,E)$ with $|A|\geq 2$ and $|B|\geq 2$,

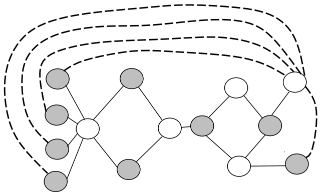


FIG. 2. *The edges of the bipartite graph in this example are drawn as the solid edges. The cardinality of an optimal biconnector of this graph is* 5. *The dashed edges form an optimal biconnector.*

find an optimal biconnector of G. An example is illustrated in Figure 2.

The remainder of this paper assumes $|A| \geq 2$ and $|B| \geq 2$. Also, let n and m be the numbers of vertices and edges in G, respectively.

Given an edge subset E' and a vertex subset V' of G, $G - V'$ denotes G without the vertices in V' and their adjacent edges. $G - E'$ denotes $(A, B, E - E')$, i.e., the resulting G after the edges in E' are deleted. $G \cup E'$ denotes $(A, B, E \cup E')$, i.e., the resulting G after the edges in E' are added to G.

2.2. Basic definitions. A *cut vertex* or *edge* of a graph is one whose removal increases the number of connected components. A *singular* connected component is one formed by an isolated vertex. A *singular* block is one with exactly one vertex. An *isolated* block is one that is also a connected component. A *pendant* block is a singular block consisting of a vertex of degree 1 or a nonsingular block containing exactly one cut vertex. Let $\Lambda(G)$ denote the set of pendant blocks of G.

The *block tree* of G is a tree $\Psi(G)$ defined as follows. D_1 denotes the set of nonsingular blocks of G. D_2 is that of singular pendant ones. D_3 is that of singular nonpendant ones. C is that of cut vertices. K is that of cut edges. The vertex set of $\Psi(G)$ is $D_1 \cup D_2 \cup C \cup K$, where D_3 is excluded because if $\{u\} \in D_3$, then $u \in C$. The vertices in $\Psi(G)$ corresponding to $D_1 \cup D_2$ are called the *b-vertices*; those corresponding to $C \cup K$ are the *c-vertices*. To distinguish between an edge in G and one in $\Psi(G)$, let $\langle y_1, y_2 \rangle$ instead of (y_1, y_2) denote an edge between two vertices y_1 and y_2 in $\Psi(G)$. The edge set of $\Psi(G)$ is the union of the following sets:

- $\{\langle d_1, c \rangle \mid d_1 \in D_1$ and $c \in C$ such that $c \in D_1\}$;
- $\{\langle c, e \rangle \mid c \in C$ and $e \in K$ such that c is an endpoint of $e\}$;
- $\{\langle e, d_2 \rangle \mid e \in K$ and $d_2 \in D_2$ such that an endpoint of e is in $d_2\}$.

Figure 3 illustrates G and its blocks while Figure 4 illustrates its block tree.

LEMMA 2.1.

1. $\Psi(G)$ *is a tree with* $O(n)$ *vertices. Its leaves are the* $|\Lambda(G)|$ *pendant blocks of* G.
2. *For all cut vertices* u *in* G, $\mathcal{D}(u, G)$ *equals the degree of* u *in* $\Psi(G)$.

Proof. The proof is straightforward and similar to that for similar constructs [9]. □

Let $P_{u,v}$ denote the tree path between two vertices u and v in $\Psi(G)$. Let $|P_{u,v}|$ be the number of vertices in $P_{u,v}$.

LEMMA 2.2. *Let* Y_1 *and* Y_2 *be a legal pair of* G. *Let* e *be a binding edge for* Y_1 *and* Y_2. *Let* $G' = G \cup \{e\}$.

1. *The cut vertices of* G *corresponding to c-vertices in* P_{Y_1,Y_2} *and the vertices of* G *in the b-vertices on* P_{Y_1,Y_2} *form a new block* Y_e *in* G'. *The b-vertices of* $\Psi(G')$ *are* Y_e *and those of* $\Psi(G)$ *not on* P_{Y_1,Y_2}.

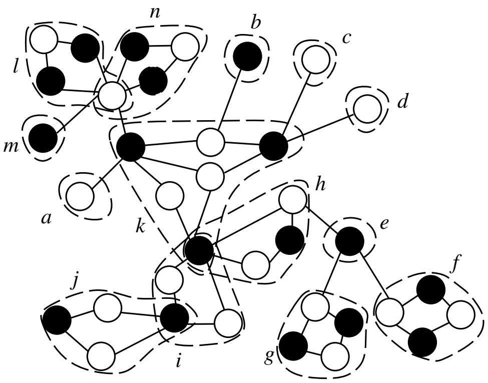


FIG. 3. *In this bipartite graph $G = (A, B, E)$, A is the set of shaded vertices, and B the set of unshaded vertices. The vertices in each block of G are grouped into a dashed circle.*

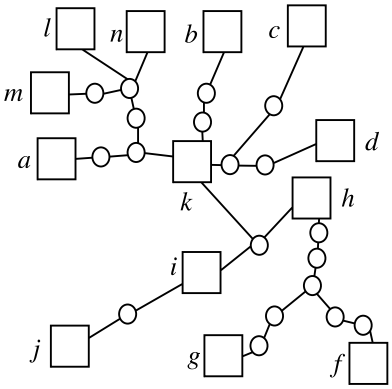


FIG. 4. *This is the block tree of the graph in Figure 3. The boxes are the latter's nonsingular blocks and singular pendant ones; the circles are its cut edges and cut vertices.*

2. *The c-vertices in $\Psi(G')$ are those in $\Psi(G)$ excluding the ones on P_{Y_1,Y_2} that are of degree two in $\Psi(G)$.*
3. *The edge set of $\Psi(G')$ is the union of*
   - *the set of edges in $\Psi(G)$ whose two endpoints are still in $\Psi(G')$;*
   - $\{\langle u, Y_e\rangle \mid u \in P_{Y_1,Y_2}$ *is a cut vertex of G that remains in $\Psi(G')\}$;*
   - $\{\langle u, Y_e\rangle \mid u \notin P_{Y_1,Y_2}$ *is a cut vertex of G incident to P_{Y_1,Y_2} in $\Psi(G)\}$.*
4. *The number of vertices in $\Psi(G')$ is at most that for $\Psi(G)$ minus $\frac{|P_{Y_1,Y_2}|-1}{2}$.*
5. *If P_{Y_1,Y_2} contains a b-vertex of degree at least four in $\Psi(G)$ or two vertices each of degree at least three, then $\Lambda(G') = \Lambda(G) - \{Y_1, Y_2\}$.*

Proof. The proof is straightforward and similar to those for similar constructs [5, 9, 20, 29]. □

A vertex of G is *type* A or B if it is in A or B, respectively. A block of G is *type* A or B if all of its noncut vertices are in A or B, respectively; a block is *type* AB if it has at least one noncut vertex in A and one in B. A *legal pair* of G is formed by two distinct elements in $\Lambda(G)$ paired according to the following rules. Type A may pair with type B or AB. Type B may pair with type A or AB. Type AB may pair with all three types. A *binding* edge for a legal pair is a legal edge between two noncut vertices, one from each of the two blocks of the pair.

LEMMA 2.3.

1. *A noncut vertex is in exactly one block. Each pendant block contains a noncut vertex.*
2. *A singular pendant block of G is either type A or B, while a nonsingular pendant block is type AB and has at least two vertices from A and at least two from B.*
3. *There exists a binding edge for each legal pair of G.*

Proof. The first two statements are straightforward. We outline the proof of the third statement. Let G_1 and G_2 be a legal pair of G. There are three cases depending on the type of G_1.

Case 1: G_1 is type A. Then G_2 is either type B or type AB. By the first two statements, there is a vertex $u \in A$ such that $u \in G_1$ and $u \notin G_2$. There is also a vertex $v \in B$ such that $v \in G_2$ and $v \notin G_1$. The edge (u, v) is a binding edge.

Case 2: G_1 is type B. Then G_2 is either type A or type AB. The proof is similar to that of Case 1.

Case 3: G_1 is type AB. Then G_2 can be any type. The proof is similar to that of Case 1. □

Let $\Lambda' \subseteq \Lambda(G)$. A *legal matching* of Λ' is a set of legal pairs between elements in Λ' such that each element in Λ' is in at most one legal pair. A *maximum* legal matching of Λ' is one with the largest cardinality possible. $\mathcal{M}(\Lambda')$ denotes the cardinality of a maximum legal matching of Λ'. For a maximum legal matching of Λ', let

$$\mathcal{R}(\Lambda') = |\Lambda'| - 2\mathcal{M}(\Lambda'),$$

i.e., the number of elements in Λ' that are not in the given maximum legal matching. Note that $\mathcal{R}(\Lambda')$ is the same for any maximum legal matching of Λ'.

LEMMA 2.4.

1. *Let W_1 and W_2 be two disjoint nonempty sets of pendant blocks with $\mathcal{M}(W_1 \cup W_2) > 0$. Then some $w_1 \in W_1$ and $w_2 \in W_2$ form a legal pair with $\mathcal{M}(W_1 \cup W_2 - \{w_1, w_2\}) = \mathcal{M}(W_1 \cup W_2) - 1$.*
2. *Let n_A, n_B, and n_{AB} be the numbers of type A, B, and AB pendant blocks in $\Lambda(G)$, respectively. Then, $\mathcal{R}(\Lambda(G)) = n_A + n_B + n_{AB} - 2\mathcal{M}(\Lambda(G))$ and $\mathcal{M}(\Lambda(G)) = \alpha + \beta + \gamma$, where $\alpha = \min\{n_A, n_B\}$, $\beta = \min\{|n_A - n_B|, n_{AB}\}$ and $\gamma = \lfloor \frac{n_{AB}-\beta}{2} \rfloor$.*

Proof. The first statement follows from the fact that $W_1 \cup W_2$ has a maximum legal matching that contains a legal pair between W_1 and W_2. The second statement follows from the fact that a legal matching can be obtained by iteratively applying any applicable rule below.

- If there are one unpaired type A pendant block and one unpaired type B pendant block, then we pair a type A pendant block and a type B one.
- If there is no unpaired type B (respectively, A) pendant block and there are one unpaired type A (respectively, B) pendant block and one unpaired type AB pendant block, then we pair a type A (respectively, B) pendant block with a type AB one.
- If all unpaired pendant blocks are type AB, then we pair two such blocks.

We now prove that these rules produce a maximum matching. Assume we run the above process, and let $\Lambda^*(G)$ be the set of pendant blocks that is not in the matching produced. Note that $\Lambda^*(G)$ consists of pendant blocks of the same type, since any two pendant blocks of different types can be matched. There are three cases.

Case 1: $\Lambda^*(G)$ consists of only type AB pendant blocks. Note that two type AB pendant blocks can be matched. Therefore, $|\Lambda^*(G)| = 1$. Thus $|\Lambda(G)|$ is odd and we have produced a maximum matching.

Case 2: $\Lambda^*(G)$ consists of only type A pendant blocks. From our matching rules, a type A pendant block is matched with a type B or AB pendant block whenever possible. Thus $|\Lambda^*(G)| = n_A - n_B - n_{AB}$. Since type A pendant blocks can only be matched with type B or AB pendant blocks, we have produced a maximum matching.

Case 3: $\Lambda^*(G)$ consists of only type B pendant blocks. This case is similar to Case 2. □

For all vertices $u \in G$, $\mathcal{D}(u, G)$ denotes the number of connected components in $X - \{u\}$ where X is the connected component of G containing u. $\mathcal{C}(G)$ denotes the number of connected components in G that are not blocks. $\mathcal{B}(G)$ denotes the number of edges in an optimal biconnector of G. When G is connected, our target size for an optimal biconnector is

$$\eta(G) = \max_{u \in G}\{\mathcal{D}(u, G) + \mathcal{C}(G) - 2, \mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G))\}.$$

2.3. Main results. We first prove a lower bound on the size of an optimal biconnector and then discuss two main results of this paper.

Lemma 2.5.

1. *G is componentwise biconnected if and only if $\eta(G) = 0$.*
2. *$\mathcal{B}(G) \geq \eta(G)$.*

Proof. Statement 1. If G is componentwise biconnected, then $\Lambda(G) = \emptyset$, $\mathcal{C}(G) = 0$, and $\mathcal{D}(u, G) = 1$ for all vertices $u \in G$. Hence $\eta(G) = 0$. We next prove the only-if direction. Since $\eta(G) = 0$, $\mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G)) = 0$. Then, since $\mathcal{M}(\Lambda(G)), \mathcal{R}(\Lambda(G)) \geq 0$ by definition, $\mathcal{M}(\Lambda(G)) = \mathcal{R}(\Lambda(G)) = 0$, and $\Lambda(G) = \emptyset$. By Lemma 2.1(1), $\Psi(G^*)$ is a tree for each connected component G^* in G. The leaves of $\Psi(G^*)$ are in $\Lambda(G)$. Hence $\Psi(G^*)$ is a one-vertex tree. Then G^* is biconnected, implying that G is componentwise biconnected.

Statement 2. It suffices to show $\mathcal{B}(G) \geq \mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G))$ and $\mathcal{B}(G) \geq \max_{u \in G} \mathcal{D}(u, G) + \mathcal{C}(G) - 2$. Let L be an optimal biconnector of G.

To prove $\mathcal{B}(G) \geq \mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G))$, note that $\Lambda(G \cup L)$ is empty. Thus, every block in $\Lambda(G)$ contains an endpoint of an edge in L. Since all the edges in L are legal, L contains at least $\mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G))$ edges.

To prove $\mathcal{B}(G) \geq \max_{u \in G} \mathcal{D}(u, G) + \mathcal{C}(G) - 2$, we need such an L that the nonblock connected components of G are all contained in the same connected component of $G \cup L$. If a given L has not yet satisfied this property, then let X_1 and X_2 be two nonblock connected components of G that are contained in two different connected components X_1' and X_2' of $G \cup L$, respectively. Let $e_1 = (u_1, v_1) \in X_1'$ and $e_2 = (u_2, v_2) \in X_2'$ be two edges in L. Such e_1 and e_2 exist because X_1 and X_2 are not biconnected in G, but X_1' and X_2' are biconnected in $G \cup L$. Next, let $e_1' = (u_1, v_2)$ and $e_2' = (u_2, v_1)$. Then, $L' = (L - \{e_1, e_2\}) \cup \{e_1', e_2'\}$ remains an optimal biconnector of G. Also, L' connects $X_1' - \{e_1\}$ and $X_2' - \{e_2\}$, which include X_1 and X_2. An example is illustrated in Figure 5.

By repeating this endpoint switching process, we can construct a desired L. With such an L, we proceed to prove $\mathcal{B}(G) \geq \max_{u \in G} \mathcal{D}(u, G) + \mathcal{C}(G) - 2$. Since this claim trivially holds if G is componentwise biconnected, we focus on the case where G is not componentwise biconnected. Then, $\mathcal{D}(u, G)$ is maximized by some u that is in a nonblock connected component H_u^*. By definition, $H_u^* - \{u\}$ contains $\mathcal{D}(u, G)$ connected components. There are $\mathcal{C}(G)$ nonblock connected components, one of which

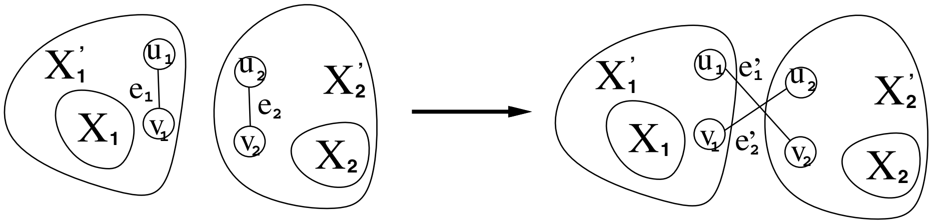


Fig. 5.

is H_u^*. Thus there are at least $\mathcal{C}(G) - 1 + \mathcal{D}(u, G)$ connected components in $G - \{u\}$. It is well known that we need to add at least k edges to connected a graph with $k+1$ connected components. Let H_u be the connected component of $G \cup L$ containing u. Since $G \cup L$ is componentwise biconnected, $H_u - \{u\}$ is connected. Thus $|L| \geq \mathcal{D}(u, G) + \mathcal{C}(G) - 2$, proving our claim. □

The next theorem is a main result of this paper.

Theorem 2.6. *If G is connected, then $\mathcal{B}(G) = \eta(G)$.*

Proof. By Lemma 2.5, $\mathcal{B}(G) \geq \eta(G)$. For ease of understanding, the proof for $\mathcal{B}(G) \leq \eta(G)$ is delayed to Theorem 3.2 in section 3. □

The next theorem generalizes Theorem 2.6 to G that may or may not be connected.

- Let $\mathcal{C}_1(G)$ be the number of connected components of G that are neither isolated edges nor blocks.
- Let $\mathcal{C}_2(G)$ be the number of isolated edges; note that $\mathcal{C}(G) = \mathcal{C}_1(G) + \mathcal{C}_2(G)$.
- Let $\mathcal{C}_3(G)$ be the number of connected components that are nonsingular blocks.

Theorem 2.7.

Case M1: $\mathcal{C}_1(G) = 1$ *and* $\mathcal{C}_2(G) = 0$. *Then* $\mathcal{B}(G) = \eta(G)$.

Case M2: $\mathcal{C}_1(G) + \mathcal{C}_2(G) \geq 2$ *and* $\mathcal{M}(\Lambda(G)) = 0$. *Then* $\mathcal{B}(G) = \eta(G)$.

Case M3: $\mathcal{C}_1(G) + \mathcal{C}_2(G) \geq 2$ *and* $\mathcal{M}(\Lambda(G)) > 0$. *Then* $\mathcal{B}(G) = \eta(G)$.

Case M4: $\mathcal{C}_1(G) = 0$, $\mathcal{C}_2(G) = 1$, *and* $\mathcal{C}_3(G) = 0$. *Then* $\mathcal{B}(G) = 3$.

Case M5: $\mathcal{C}_1(G) = 0$, $\mathcal{C}_2(G) = 1$, *and* $\mathcal{C}_3(G) > 0$. *Then* $\mathcal{B}(G) = 2$.

Case M6: $\mathcal{C}(G) = 0$. *Then* $\mathcal{B}(G) = 0$.

Proof.

Case M1: Let G_1 be the connected component of G that is neither an isolated edge nor a block. Theorem 2.6 applies to the case where G_1 contains at least two vertices in A and at least two in B. Thus, we may assume without loss of generality that G_1 contains exactly one vertex $u \in A$ and r vertices $v_1, v_2, \ldots, v_r \in B$ with $r \geq 2$. Note that $\eta(G) = r$. Because $|A| > 1$ and $\mathcal{C}_2(G) = 0$, there is an isolated vertex $w \in A$ or there is a nonsingular block in G containing two vertices $w_1, w_2 \in A$. In the former case, $\{(w, v_1), \ldots, (w, v_r)\}$ is an optimal biconnector; in the latter case, $\{(w_1, v_1)\} \cup \{(w_2, v_2), (w_2, v_3), \ldots, (w_2, v_r)\}$ is an optimal biconnector.

Case M2: Since $\mathcal{M}(\Lambda(G)) = 0$, we may assume without loss of generality that all the pendant blocks are type A. Note that an isolated edge contains two pendant blocks and that these two pendant blocks are of different types. Hence $\mathcal{C}_2(G) = 0$, $\mathcal{C}_1(G) \geq 2$ and $\eta(G) = |\Lambda(G)|$. Let $G_0, \ldots, G_{\mathcal{C}_1(G)-1}$ be the connected components of G that are neither isolated edges nor blocks. Since each G_i has more than two vertices, G_i has a vertex $y_i \in B$. Let $W_{i,1}, \ldots, W_{i,r_i}$ be the pendant blocks of G_i.

Each $W_{i,j}$ contains a noncut vertex $x_{i,j} \in A$. The set $\{(x_{i,j}, y_{i+1 \bmod \mathcal{C}_1(G)}) \mid 0 \leq i < \mathcal{C}_1(G)$ and $1 \leq j \leq r_i\}$ is a biconnector. By Lemma 2.5(2), this biconnector is optimal.

Case M3: By Lemma 2.5, $\mathcal{B}(G) \geq \eta(G)$. To prove the upper bound, let e be a legal edge of G. Let $G' = G \cup \{e\}$. We first show how to choose e so that $\eta(G') = \eta(G) - 1$. Since $\mathcal{M}(\Lambda(G)) > 0$, by Lemma 2.4(1), we can find a legal pair w_1 and w_2 in different connected components with $\mathcal{M}(\Lambda(G) - \{w_1, w_2\}) = \mathcal{M}(\Lambda(G)) - 1$. By Lemma 2.3(3), let e be a binding edge for w_1 and w_2. Note that $\Lambda(G') = \Lambda(G) - \{w_1, w_2\}$, $\mathcal{M}(\Lambda(G')) = \mathcal{M}(\Lambda(G)) - 1$, $\mathcal{R}(\Lambda(G')) = \mathcal{R}(\Lambda(G))$, and $\mathcal{M}(\Lambda(G')) + \mathcal{R}(\Lambda(G')) = \mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G)) - 1$. Note also that $\mathcal{C}(G') = \mathcal{C}(G) - 1$, and G' is the graph obtained from G by adding the edge e. Let u be a vertex in G and let H_u be the connected component containing u. By definition, $H_u - \{u\}$ contains $\mathcal{D}(u, G)$ connected components. The number of connected components in $(H_u - \{u\}) \cup \{e'\}$ is no larger than that of $H_u - \{u\}$ for any edge e'. Hence $\max_{u \in G'} \mathcal{D}(u, G') \leq \max_{u \in G} \mathcal{D}(u, G)$ and $\max_{u \in G'} \mathcal{D}(u, G') + \mathcal{C}(G') - 2 \leq \max_{u \in G} \mathcal{D}(u, G) + \mathcal{C}(G) - 2$. Thus, $\eta(G') = \eta(G) - 1$.

This process reduces $\mathcal{C}(G)$ and $\mathcal{M}(\Lambda(G))$ by 1 each. We iterate this process until either (1) $\mathcal{C}_1(G) + \mathcal{C}_2(G) = 1$ or (2) $\mathcal{C}_1(G) + \mathcal{C}_2(G) \geq 2$ and $\mathcal{M}(\Lambda(G)) = 0$. In the latter case, we use Case M2 to complete the proof. In the former case, note that we add an edge to combine two nonsingular nonbiconnected connected components into a connected component. This new connected component is neither an isolated edge nor a block. Thus, $\mathcal{C}_1(G) > 0$; i.e., $\mathcal{C}_1(G) = 1$ and $\mathcal{C}_2(G) = 0$ in the resulting G. We then use Case M1 to complete the proof of this case.

Case M4: Let (r, c) be the isolated edge. Let $r' \in A$ and $c' \in B$ be two isolated vertices. Then $\{(r, c'), (r', c), (r', c')\}$ is an optimal biconnector of G.

Case M5: Let G' be a connected component that is a nonsingular block in G. G' has a vertex $r \in A$ and a vertex $c \in B$. Let (r', c') be the isolated edge of G. Then $\{(r, c'), (r', c)\}$ is an optimal biconnector of G.

Case M6: This case is straightforward. □

3. A matching upper bound for a connected G.

This section assumes that G is connected.

A cut vertex u of G is *massive* if $\mathcal{D}(u, G) - 1 > \mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G))$; it is *critical* if $\mathcal{D}(u, G) - 1 = \mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G))$.

LEMMA 3.1. *Assume* $|\Lambda(G)| \geq 3$.

1. *G has at most two critical vertices. If it has two, then* $\mathcal{R}(\Lambda(G)) = 0$.
2. *G has at most one massive vertex. If it has one, then it has no critical vertex.*

Proof. From Lemma 2.1 and the structure of a block tree, if we remove a c-vertex u from $\Psi(G)$, then the resulting graph consists of $\mathcal{D}(u, G)$ connected components each containing a pendant block in $\Lambda(G)$. If we remove two c-vertices u and v from $\Psi(G)$, then the resulting graph contains $\mathcal{D}(u, G) - 2 + \mathcal{D}(v, G)$ connected components each containing a pendant block in $\Lambda(G)$. Thus, $|\Lambda(G)| \geq \mathcal{D}(u, G) + \mathcal{D}(v, G) - 2$. Note that pendant blocks in two distinct connected components are distinct. An example is illustrated in Figure 6. By similar arguments, if we remove three c-vertices u, v, and w, then the resulting graph has at least $\mathcal{D}(u, G) + \mathcal{D}(v, G) + \mathcal{D}(w, G) - 4$ connected components each containing a pendant block in $\Lambda(G)$. Thus, $|\Lambda(G)| \geq \mathcal{D}(u, G) + \mathcal{D}(v, G) + \mathcal{D}(w, G) - 4$.

Statement 1. To prove the first part by contradiction, assume that G has at least three critical cut vertices u_1, u_2 and u_3. From the above analysis, $\sum_{i=1}^{3} \mathcal{D}(u_i, G) - 4 \leq |\Lambda(G)|$. By definition, $\mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G)) \geq \frac{|\Lambda(G)|}{2}$. Since each u_i is critical, $\mathcal{D}(u_i, G) - 1 = \mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G)) \geq \frac{|\Lambda(G)|}{2}$. Hence $\sum_{i=1}^{3}(\mathcal{D}(u_i, G) - 1) - 1 =$

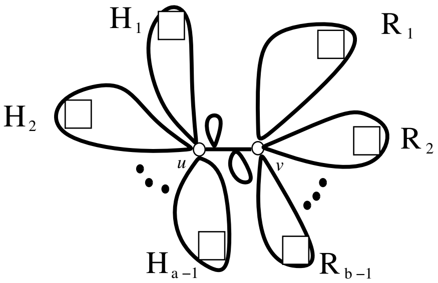


FIG. 6. *This illustrates the counting argument used in the proof of Lemma* 3.1. *The figure shows* $\Psi(G)$ *with two c-vertices* u *and* v. *Here* $a = \mathcal{D}(u, G)$ *and* $b = \mathcal{D}(v, G)$. *We can find* $a + b - 2$ *connected components, i.e.,* $H_1, \dots, H_{a-1}$ *and* $R_1, \dots, R_{b-1}$ *in* $\Psi(G) - \{u, v\}$, *each of which contains a pendant block.*

$\sum_{i=1}^{3} \mathcal{D}(u_i, G) - 4 \geq |\Lambda(G)| + \frac{|\Lambda(G)|}{2} - 1$. Since $|\Lambda(G)| \geq 3$, $\frac{|\Lambda(G)|}{2} - 1 > 0$ and $\sum_{i=1}^{3}(\mathcal{D}(u_i, G) - 1) - 1 > |\Lambda(G)|$, reaching a contradiction. Hence G cannot have more than two critical cut vertices. We now prove the second part of the statement. Assume that G has exactly two critical cut vertices u_1 and u_2. From the above analysis, $\sum_{i=1}^{2}(\mathcal{D}(u_i, G) - 1) \leq |\Lambda(G)|$. Since each u_i is critical, $\mathcal{D}(u_i, G) - 1 = \mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G))$. Hence $\sum_{i=1}^{2}(\mathcal{D}(u_i, G) - 1) = 2 \cdot (\mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G)))$. If $\mathcal{R}(\Lambda(G)) > 0$ were true, then $2 \cdot (\mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G))) > |\Lambda(G)|$, which is a contradiction. Hence, $\mathcal{R}(\Lambda(G)) = 0$.

Statement 2. To prove this statement by contradiction, let u_1 be a massive vertex, and let u_2 be a critical or massive vertex. From the above analysis, $\sum_{i=1}^{2}(\mathcal{D}(u_i, G) - 1) \leq |\Lambda(G)|$. However, $\mathcal{D}(u_1, G) - 1 > \mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G))$ since u_1 is massive, and $\mathcal{D}(u_2, G) - 1 \geq \mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G))$ since u_2 is critical or massive. Thus $\sum_{i=1}^{2}(\mathcal{D}(u_i, G) - 1) > 2 \cdot (\mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G))) \geq |\Lambda(G)|$, reaching a contradiction. Hence, this statement holds. □

The next theorem is the main result of this section.

THEOREM 3.2. $\mathcal{B}(G) \leq \eta(G)$.

Proof. By Lemma 3.1, we divide the proof into the following five cases. These cases are proved in sections 3.1–3.5, respectively.

Case S1: $|\Lambda(G)| < 3$.

Case S2: $|\Lambda(G)| \geq 3$ and $\mathcal{M}(\Lambda(G)) = 0$.

Case S3: $|\Lambda(G)| \geq 3$, $\mathcal{M}(\Lambda(G)) > 0$, and G has two critical vertices.

Case S4: $|\Lambda(G)| \geq 3$, $\mathcal{M}(\Lambda(G)) > 0$, and G has no massive vertex and at most one critical vertex.

Case S5: $|\Lambda(G)| \geq 3$, $\mathcal{M}(\Lambda(G)) > 0$, and G has exactly one massive vertex. □

3.1. Case S1 of Theorem 3.2.

LEMMA 3.3. *For Case* S1, *Theorem* 3.2 *holds. Furthermore, given* G, *an optimal biconnector can be computed in* $O(m + n)$ *time.*

Proof. By Lemma 2.1, $\Lambda(G)$ corresponds to leaves in $\Psi(G)$. If $|\Lambda(G)| = 0$, then G is componentwise biconnected and the lemma is true trivially. It is not possible that $|\Lambda(G)| = 1$. Hence assume $|\Lambda(G)| = 2$. Then $\Psi(G)$ is a single path, implying that the degree of every c-vertex in $\Psi(G)$ is 2. Hence $\mathcal{D}(u, G) + \mathcal{C}(G) - 2 = 0$ and $\eta(G) = \mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G))$. There are two cases.

Case 1: $\mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G)) = 1$. The two pendant blocks in $\Lambda(G)$ are a legal pair. The optimal biconnector consists of the legal edge e between the legal pair. It

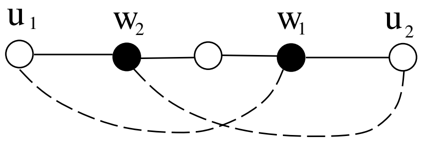


FIG. 7. *This illustrates the proof of Lemma* 3.3. *The two dotted edges form a biconnector.*

is clear that $G \cup \{e\}$ is biconnected.

Case 2: $\mathcal{M}(\Lambda(G))+\mathcal{R}(\Lambda(G)) = 2$. The two pendant blocks in $\Lambda(G)$ are of the same type. Without loss of generality, assume they are of type A. Let $u_1 \in A$ be a noncut vertex from one pendant block. Similarly let $u_2 \in A$ be a noncut vertex from the other pendant block. Since we assume $|A|, |B| \geq 2$, we can find two vertices $w_1 \in B$ and $w_2 \in B$. If w_1 and w_2 are in the same block, then $\{(w_1, u_1), (w_2, u_2)\}$ is a biconnector for G. If w_1 and w_2 are in different blocks, then assume the distance between w_1 and u_1 is longer than the distance between w_2 and u_1. Then $\{(w_1, u_1), (w_2, u_2)\}$ is a biconnector for G. An example is illustrated in Figure 7. □

3.2. Case S2 of Theorem 3.2.

LEMMA 3.4. *Theorem* 3.2 *holds for Case* S2.

Proof. Let $k = |\Lambda(G)|$. Since $\mathcal{M}(\Lambda(G)) = 0$, $\eta(G) = k$ by Lemma 2.1. It suffices to construct a biconnector L of k edges for G. Let $Y_1, \ldots, Y_k$ be the pendant blocks of G. Since $\mathcal{M}(\Lambda(G)) = 0$, $Y_i = \{y_i\}$ and we may assume $y_i \in B$ without loss of generality. Then G has a cut edge (x_i, y_i) for each y_i, where $x_i \in A$. Since $|A| \geq 2$ and $\mathcal{M}(\Lambda(G)) = 0$, there is some $x_j \neq x_1$. Let G' be the connected component of $G - \{x_1\}$ containing x_j. Let L be the set of legal edges (y_i, x_1) for all $y_i \in G'$ and (y_i, x_j) for all $y_i \notin G'$. It is straightforward to prove that L is as desired by means of Lemma 2.2. □

3.3. Case S3 of Theorem 3.2. A path $v_1, \ldots, v_k$ in $\Psi(G)$ is *branchless* if for all i with $1 < i < k$ the degree of v_i in $\Psi(G)$ is two. Let u_1 and u_2 be the critical vertices of G. A leaf *clings* to u_i in $\Psi(G)$ if there is a branchless path between it and u_i.

LEMMA 3.5.

1. $\eta(G) = \mathcal{M}(\Lambda(G)) = \frac{|\Lambda(G)|}{2}$.
2. $\Psi(G)$ *has a branchless path between* u_1 *and* u_2*, and exactly* $\frac{|\Lambda(G)|}{2}$ *leaves cling to* u_1 *only while the other* $\frac{|\Lambda(G)|}{2}$ *leaves cling to* u_2 *only.*
3. $\Lambda(G)$ *has a maximum legal matching in which each legal pair is between one clinging to* u_1 *and one clinging to* u_2.

Proof.

Statement 1. This statement follows from Lemma 3.1.

Statement 2. From Statement 1, $\mathcal{M}(\Lambda(G)) = \frac{|\Lambda(G)|}{2}$. Hence $|\Lambda(G)|$ is even. By a counting argument similar to the one used in the proof of Lemma 3.1(2), there are at least $2\cdot\mathcal{M}(\Lambda(G)) = |\Lambda(G)|$ connected components in $G - \{u_1, u_2\}$, each of which contains a pendant block. Furthermore, none of these connected components contains a vertex in the path P_{u_1,u_2}, and each of these connected components contains exactly one pendant block. Hence each pendant block clings to either u_1 or u_2, and the path P_{u_1,u_2} is branchless. An example is illustrated in Figure 8.

Statement 3. For $1 \leq i \leq 2$, let W_i be the set of pendant blocks clinging to u_i. We now prove $\Lambda(G)$ has a perfect matching in which each legal pair consists of one

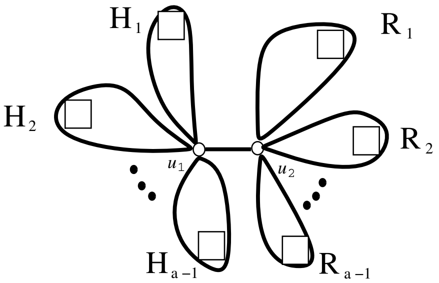


Fig. 8. *This illustrates the counting argument used in the proof of Lemma* 3.5. *The figure shows* $\Psi(G)$ *with two c-vertices* u_1 *and* u_2. *Here* $a - 1 = \mathcal{M}(\Lambda(G))$. *We can find* $2{\cdot}a - 2$ *connected components, i.e.,* $H_1, \ldots, H_{a-1}$ *and* $R_1, \ldots, R_{a-1}$ *in* $\Psi(G) - \{u_1, u_2\}$, *each of which contains a pendant block.*

pendant block from W_1 and one from W_2. Note that it is easy to see that $\Lambda(G)$ has at least one perfect matching. If any perfect matching of $\Lambda(G)$ does not have the desired property, then we iteratively modify it as follows until it does.

Assume without loss of generality that the given perfect matching contains a legal pair $t = (t_1, t_2)$, both clinging to the same critical vertex u_1. Then, there must exist another legal pair $r = (r_1, r_2)$ in the perfect matching that both cling to u_2. We replace these two pairs with two new legal pairs as follows. From the matching procedure applied in the proof of Lemma 2.4, each legal pair either (1) contains a type AB pendant block or (2) consists of a type A block and a type B block. There are four cases.

Case 1: t satisfies (1) and r satisfies (1). Without loss of generality, assume t_1 and r_1 are type AB. Hence we form the new legal pairs (t_1, r_2) and (r_1, t_2).

Case 2: t satisfies (1) and r satisfies (2). Without loss of generality, assume t_1 is type AB. We can always find one pendant block in r to match with t_2. The pendant block left in r can always match with t_1.

Case 3: t satisfies (2) and r satisfies (1). This case is similar to Case 2.

Case 4: t satisfies (2) and r satisfies (2). Without loss of generality, assume t_1 and r_1 are type A. Hence we form the new legal pairs (t_1, r_2) and (r_1, t_2). □

Lemma 3.6. *Theorem* 3.2 *holds for Case* S3.

Proof. We add to G a binding edge for each legal pair in the maximum legal matching of Lemma 3.5. By Lemmas 2.2 and 3.5, the resulting graph is biconnected. We add $\frac{|\Lambda(G)|}{2}$ edges, which by Lemma 3.5(1) is optimal. □

3.4. Case S4 of Theorem 3.2. Since $|\Lambda(G)| \geq 3$, we can divide Case S4 into two subcases:

Case S4-1: $\Psi(G)$ has exactly one vertex of degree at least three.

Case S4-2: $\Psi(G)$ has more than one vertex of degree at least three.

Lemma 3.7. *Theorem* 3.2 *holds for Case* S4-1.

Proof. Let x be the vertex in $\Psi(G)$ of degree at least three. There are two cases:

Case 1: x is a *b*-vertex. Then $\eta(G) = \mathcal{R}(\Lambda(G)) + \mathcal{M}(\Lambda(G))$.

Case 2: x is a *c*-vertex. Since x is not massive, $\eta(G) = \mathcal{D}(x, G) - 1 = \mathcal{R}(\Lambda(G)) + \mathcal{M}(\Lambda(G))$ and $\mathcal{M}(\Lambda(G)) = 1$.

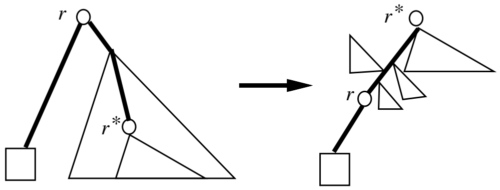


Fig. 9. *This illustrates Case* 2 *in the proof of Lemma* 3.8. *The original* $\Psi(G)$ *is shown on the left. The rerooted block tree is shown on the right. The leftmost branch of the original* $\Psi(G)$ *is a branch. The degree of* r^* *is at least three.*

In either case, let N_1 be a maximum legal matching of $\Lambda(G)$; next, let N_2 be a set of legal pairs formed by pairing each pendant block not yet matched in N_1 with one already matched. Then, $N = N_1 \cup N_2$ is a set of the smallest number of legal pairs of G such that each element in $\Lambda(G)$ is in a pair. We add to G a binding edge for each pair in N. Since $\mathcal{M}(\Lambda(G)) > 0$, we add $\eta(G)$ edges. Since $\mathcal{M}(\Lambda(G)) > 0$ in Case 1 and $\mathcal{M}(\Lambda(G)) = 1$ in Case 2, these edges form a desired biconnector by Lemma 2.2. □

To discuss Case S4-2, we further assume that $\Psi(G)$ is rooted at a vertex with at least two neighbors; however, the degree of a vertex in $\Psi(G)$ still refers to its number of neighbors instead of children.

The next lemma chooses an advantageous root for $\Psi(G)$ for our augmentation algorithm. Given a vertex v in $\Psi(G)$, a *branch* of v, also called a *v-branch*, is the subtree of $\Psi(G)$ rooted at a child of v. A *chain* of v, also called a *v-chain*, is a v-branch that contains exactly one leaf in $\Psi(G)$.

Let c^* be a c-vertex in $\Psi(G)$ of the largest possible degree.

Lemma 3.8. *In Case* S4-2, *we can reroot* $\Psi(G)$ *at a vertex* h *such that*

1. *either* h *is of degree two and no* h*-branch is a chain, or* h *is of degree at least three;*
2. *if* c^* *is critical, then* $h = c^*$.

Proof. Let r be the current root of $\Psi(G)$. There are three cases.

Case 1: c^* is not critical, and either r is of degree two and no r-branch is a chain or r is of degree at least three. We set $h = r$.

Case 2: c^* is not critical, r is of degree two, and an r-branch is a chain. Note that $\Psi(G)$ has a vertex r^* of degree at least three. We set $h = r^*$. An example is illustrated in Figure 9.

Case 3: c^* is critical. Since $|\Lambda(G)| \geq 3$, c^* is of degree three or more. We set $h = c^*$. □

Lemma 3.9. *Let* h *be the root of* $\Psi(G)$. *In Case* S4-2, *if* h *is of degree two and no* h*-branch is a chain or if* h *is of degree at least three, then* G *has a legal pair* w_1 *and* w_2 *such that*

1. P_{w_1,w_2} *passes through* h *and two vertices of degree at least three;*
2. $\mathcal{M}(\Lambda(G) - \{w_1, w_2\}) = \mathcal{M}(\Lambda(G)) - 1$.

Proof. There are two cases.

Case 1: The degree of h is two and no h-branch is a chain. Let T^* be an h-branch.

Case 2: The degree of h is at least three. Since this is Case S4-2, some descendant of h has degree at least three. Let T^* be the h-branch containing that descendant.

Let W_1 be the set of leaves in T^*. Let $W_2 = \Lambda(G) - W_1$. By Lemma 2.4(1), there exist a legal pair $w_1 \in W_1$ and $w_2 \in W_2$ with $\mathcal{M}(\Lambda(G) - \{w_1, w_2\}) = \mathcal{M}(\Lambda(G)) - 1$. Then, P_{w_1,w_2} contains h as desired. An example is illustrated in Figure 10.

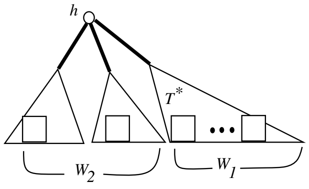


FIG. 10. *An iIllustration of Case* 2 *in the proof of Lemma* 3.9. *Any path between a leaf in* W_1 *and one in* W_2 *must pass through the root* h.

Furthermore, in Case 1, P_{w_1,w_2} contains a vertex of degree at least three in T^* and another in $\Psi(G) - T^*$; in Case 2, h itself is of degree at least three, and P_{w_1,w_2} contains a vertex of degree at least three in T^*. In both cases, P_{w_1,w_2} is as desired. □

LEMMA 3.10. *In Case* S4-2*, we can add a legal edge to* G *such that*

1. *the resulting graph* G' *satisfies Case* S1*,* S2*,* S3*, or* S4*;*
2. $\eta(G') = \eta(G) - 1$*;*
3. *if* G *has a critical vertex, then that vertex remains critical in* G'.

Proof. We use Lemma 3.8 to reroot $\Psi(G)$, use Lemma 3.9 to pick a legal pair w_1 and w_2, and then add a binding edge for this pair to G. By Lemmas 3.9(1) and 2.2(5), $\Lambda(G') = \Lambda(G) - \{w_1, w_2\}$. By Lemma 3.9(2), $\mathcal{M}(\Lambda(G')) = \mathcal{M}(\Lambda(G)) - 1$. Hence $\mathcal{M}(\Lambda(G')) + \mathcal{R}(\Lambda(G')) = \mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G)) - 1$. There are two cases.

Case 1: G has no critical vertex. Then, by Lemma 2.2, $\max_{u\in G'} \mathcal{D}(u, G') \leq \max_{u\in G} \mathcal{D}(u, G) \leq \mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G))$.

Case 2: G has a critical vertex. Then, c^* is the critical vertex and $\mathcal{D}(c^*, G) > \max_{u\neq c^*} \mathcal{D}(u, G)$. By Lemmas 3.8(2), 3.9(1), and 2.2, $\max_{u\in G'} \mathcal{D}(u, G') = \max_{u\in G} \mathcal{D}(u, G) - 1 = \mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G))$. Hence c^* remains to be a critical vertex.

In either case, $\max_{u\in G'} \mathcal{D}(u, G') - 1 \leq \mathcal{M}(\Lambda(G')) + \mathcal{R}(\Lambda(G'))$. Then, $\eta(G') = \eta(G) - 1$. Also, G' has no massive vertex and thus satisfies Case S1, S2, S3, or S4. □

LEMMA 3.11. *Theorem* 3.2 *holds for Case* S4.

Proof. For Case S4-1, we use Lemma 3.7. For Case S4-2, we add one edge to G at a time using Lemma 3.10 until the resulting graph G' does not satisfy Case S4-2. By Lemma 3.10(1), G' satisfies Case S1, S2, S3, or S4-1. Thus, we apply Lemma 3.3, 3.4, 3.6, or 3.7 to G' accordingly. By Lemma 3.10(2), the number of edges added is $\eta(G)$. □

3.5. Case S5 of Theorem 3.2.

Let r be the massive cut vertex of G. Let $\Psi(G)$ be rooted at r.

LEMMA 3.12.

1. $\eta(G) = \mathcal{D}(r, G) - 1 > \mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G)) > \mathcal{D}(u, G) - 1$ *for any vertex* $u \neq r$.
2. $\mathcal{D}(r, G) \geq 4$ *and there are at least four* r*-chains.*
3. *The tree* $\Psi(G)$ *contains a legal pair* Y_1 *and* Y_2 *as well as two distinct* r*-branches* T_1 *and* T_2 *such that* T_1 *is a chain,* $Y_1 \in T_1$*, and* $Y_2 \in T_2$.

Proof.

Statement 1. This statement follows from the definition of Case S5.

Statement 2. Let δ_1 be the number of r-chains. Then, $\mathcal{D}(r, G) \geq \delta_1$ and $|\Lambda(G)| \geq 2(\mathcal{D}(r, G) - \delta_1) + \delta_1$. So $\mathcal{D}(r, G) \geq \delta_1 \geq 2\mathcal{D}(r, G) - |\Lambda(G)|$. Let

$\delta_2 = (\mathcal{D}(r,G)-1) - (\mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G)))$. Because r is massive, $\delta_2 \geq 1$. Note that $|\Lambda(G)| = 2\mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G))$. Thus $\mathcal{D}(r,G) \geq \delta_1 \geq 2\delta_2 + 2 + \mathcal{R}(\Lambda(G)) \geq 4$.

Statement 3. Let T_1 be an r-chain. Let Y_1 be the leaf of $\Psi(G)$ in T_1. Because $\mathcal{M}(\Lambda(G)) > 0$, $\Psi(G)$ contains a leaf $Y_2 \neq Y_1$ that forms a legal pair with Y_1. Let T_2 be the r-branch that contains Y_2. Then, Y_1, Y_2, T_1, and T_2 are as desired. □

Lemma 3.13. *We can add a legal edge to G such that for the resulting graph G',*

1. $\eta(G') = \eta(G) - 1$*;*
2. $\mathcal{D}(r,G') = \mathcal{D}(r,G) - 1$.

Proof. Let Y_1, Y_2, T_1, and T_2 be as stated in Lemma 3.12(3). The added edge is a binding edge for Y_1 and Y_2. By Lemma 2.2, the b-vertices and c-vertices on P_{Y_1,Y_2} form a new block Y' in G'. Y' may or may not be a leaf in $\Psi(G')$; in either case, $\mathcal{M}(\Lambda(G')) + \mathcal{R}(\Lambda(G')) \leq \mathcal{M}(\Lambda(G)) + \mathcal{R}(\Lambda(G))$. Note that P_{Y_1,Y_2} contains r. Thus, by Lemmas 2.2 and 3.12(2), r remains a cut vertex in G' with $\mathcal{D}(r,G') = \mathcal{D}(r,G) - 1$ while $\mathcal{D}(v,G') \leq \mathcal{D}(v,G)$ for all vertices $v \neq r$. Consequently, $\eta(G') = \eta(G) - 1$. □

Lemma 3.14. *Theorem* 3.2 *holds for Case* S5.

Proof. We add one edge to G at a time using Lemma 3.13 until the resulting graph G' satisfies Case S1, S2, S3 or S4. Thus, we apply Lemma 3.3, 3.4, 3.6, or 3.11 accordingly. By Lemma 3.13(1), $\eta(G)$ edges are added. □

4. Computing an optimal biconnector in linear time.

Lemma 4.1. *Case* S5 *in Theorem* 3.2 *can be reduced in linear time to Case* S1*,* S2*,* S3 *or* S4.

Proof. To implement the reduction, we first define a data structure as follows. Let Q be the set of leaves of $\Psi(G)$ that are in the r-chains. We set up a counter for the number of these leaves. We also set up three doubly linked lists containing those that are types A, B, and AB, respectively.

We set up a counter for the number of r-branches that are not chains. For each such branch, we set up a doubly linked list for the leaves of $\Psi(G)$ in it. We also set up three doubly linked lists for the leaves in these branches that are types A, B, and AB, respectively.

Given G, we can set up these linked lists and counters in linear time. We next use this data structure to find a legal pair Y_1 and Y_2 by means of Lemma 3.12(3). Since $|Q| \geq 4$ by Lemma 3.12(2), there are two cases.

Case 1: Some Y_1 and $Y_2 \in Q$ form a legal pair. This is our desired pair. Note the r-chains containing Y_1 and Y_2 in $\Psi(G)$ are contracted into a new chain in $\Psi(G')$ consisting of a single leaf of type AB.

Case 2: Q contains only type A or B leaves. Select any $Y_1 \in Q$. Since $\mathcal{M}(\Lambda(G)) > 0$, some $Y_2 \in \Lambda(G) - Q$ forms a desired legal pair with Y_1. Note that Y_1 and Y_2 are no longer pendant blocks in G' and the newly created block is not a pendant block of G', either. The r-branch containing Y_2 becomes a chain if in G it contains exactly two pendant blocks.

It takes $O(1)$ time to decide which of these two cases holds. In either case, the selection of Y_1 and Y_2 takes $O(1)$ time using the linked lists. Once Y_1 and Y_2 are found, we can find a binding edge in $O(1)$ time in a straightforward manner. After the edge is added to G, we can update the data structure in $O(1)$ time for G'. Then we use Lemma 2.4(2) and the counters to check whether G' satisfies Case S5 in $O(1)$ time. We repeat this process until G' does not satisfy Case S5. At this point, we complete the reduction. Since we iteratively add at most $O(n)$ edges in Case S5, the reduction takes linear time. □

THEOREM 4.2. *Given G, an optimal biconnector is computable in $O(m+n)$ time.*

We prove this theorem by means of Theorems 2.7 and 3.2 as follows.

Given G, it takes $O(m+n)$ time to determine which case of Theorem 2.7 holds. Then it takes $O(m+n)$ time in a straightforward manner to compute an optimal biconnector for Cases M2, M4, M5, and M6; reduce Case M3 to Case M1 or M2; and reduce Case M1 to Theorem 3.2.

Next, it takes $O(m+n)$ time to determine which case of Theorem 3.2 holds. Then, it is straightforward to compute an optimal biconnector in $O(m+n)$ time for Cases S1, S2, and S3. Lemma 4.1 reduces Case S5 in $O(m+n)$ time to Case S1, S2, S3, or S4. By Lemma 3.7, we can find an optimal biconnector in $O(m+n)$ time for Case S4-1. The remaining proof shows how to reduce Case S4-2 to Case S1, S2, S3, or S4-1 in $O(m+n)$ time by implementing the proof of Lemma 3.11.

We define a data structure $\Delta(G)$ as follows. First, we root $\Psi(G)$ at a vertex of degree two or more as in section 3.4 and classify each vertex u by a 4-bit code $\sigma_0\sigma_1\sigma_2\sigma_3$ based on the subtree T_u of $\Psi(G)$ rooted at u:

- $\sigma_0 = 1$ if and only if T_u has more than one leaf;
- σ_1, σ_2, or $\sigma_3 = 1$ if and only if T_u contains a leaf of type A, B or AB, respectively.

The code has at most ten combinations, i.e., 0100, 0010, 0001, and all the combinations with $\sigma_0 = 1$ except 1000. $\Delta(G)$ is $\Psi(G)$ augmented with the following items:

1. At each vertex in $\Psi(G)$, $\Delta(G)$ maintains its degree and a doubly linked list for the children of u with the same $\sigma_0\sigma_1\sigma_2\sigma_3$ code. There are 10 such lists.
2. There are three counters for the numbers of leaves in $\Psi(G)$ of types A, B, and AB, respectively.
3. The c-vertices of degree at least three are partitioned into groups of the same degree. Each nonempty group is arranged into a doubly linked list. The lists themselves are connected by a doubly linked list in the increasing order of vertex degrees.

We do not need parent pointers in $\Delta(G)$, which are subtle to update [19, 20, 29]. This finishes the description of $\Delta(G)$. We can build $\Delta(G)$ from G in $O(m+n)$ time.

LEMMA 4.3.

1. *Let r be the current root of $\Delta(G)$. Let h be as stated in Lemma* 3.8. *Given $\Delta(G)$, if r is critical, we can reroot $\Delta(G)$ in $O(1)$ time according to Lemma* 3.8*; $O(n)$ time if r is not critical but h is; or $O(|P_{r,h}|)$ time if neither is critical.*
2. *Given $\Delta(G)$, we can find w_1 and w_2 of Lemma* 3.9 *in $O(|P_{w_1,w_2}|)$ time.*

Proof.

Statement 1. We implement the proof of Lemma 3.8 using the following steps.

Step 1. Use item 3 of $\Delta(G)$ to find c^*.

Step 2. Use items 1 and 2 of $\Delta(G)$ and Lemma 2.4(2) to decide which case of the proof of Lemma 3.8 holds.

Step 3. (a) For Case 1 of the proof of Lemma 3.8, set $h = r$ and $\Delta(G)$ is unchanged.
(b) For Case 2 of the proof of Lemma 3.8, first use item 1 of $\Delta(G)$ to find the nearest desired descendant r^* of r, and then reroot $\Delta(G)$ at $h = r^*$ and update it accordingly.
(c) For Case 3 of the proof of Lemma 3.8, if $r \neq c^*$, then recompute $\Delta(G)$ from $\Psi(G)$ to root at $h = c^*$; otherwise, $r = c^*$, and $\Delta(G)$ is unchanged.

Since Steps 1 and 2 take $O(1)$ time, the time complexity of each case of this statement is bounded by that of Step 3.

Case 1: r is critical. Step 3(c) runs with $r = c^*$ in $O(1)$ time.

Case 2: r is not critical but h is. Step 3(c) runs with $r \neq c^*$ in $O(n)$ time.

Case 3: Neither r nor h is critical. Then, Step 3(a) or Step 3(b) is performed. Step 3(a) takes $O(1)$ time. For Step 3(b), the search for r^* takes $O(1)$ time per vertex on P_{r,r^*}. Since the internal vertices of P_{r,r^*} all have degree two, updating item 1 of $\Delta(G)$ along this path takes $O(1)$ time per vertex. Item 1 of $\Delta(G)$ outside this path and the other two items remain the same. Thus, this case takes $O(P_{r,h})$ total time as desired.

Statement 2. We implement the proof of Lemma 3.9 using the following steps.

Step 1. Use item 1 of $\Delta(G)$ to decide which case of the proof of Lemma 3.9 holds.

Step 2. Use item 2 of $\Delta(G)$ and Lemma 2.4(2) to find all possible pairs of types t_1 and t_2 such that $\Lambda(G)$ has a maximum matching that contains a legal pair between type t_1 and type t_2.

Step 3. For each such pair of t_1 and t_2, perform the following computation until w_1 and w_2 are found.

(a) For Case 1 of the proof of Lemma 3.9, w_1 and w_2 are in the two branches of the root of $\Delta(G)$ separately. Use item 1 of $\Delta(G)$ at the root to decide whether the desired w_1 and w_2 exist. If they exist, use item 1 of $\Delta(G)$ to search for them.

(b) For Case 2 of the proof of Lemma 3.9, w_1 and w_2 are in two separate branches of the root, one of which is not a chain. The remaining computation is similar to that of Step 3(a).

By Lemma 3.8, some pair t_1 and t_2 yields the desired w_1 and w_2. Steps 1 and 2 take $O(1)$ time. There are $O(1)$ possible pairs of t_1 and t_2. For each such pair, checking the existence of w_1 and w_2 takes $O(1)$ time. If they exist, searching for them takes $O(1)$ time per vertex on the path P_{w_1,w_2}. □

The next lemma completes the proof of Theorem 4.2.

LEMMA 4.4. *Case* S4-2 *is reducible to Case* S1*,* S2*,* S3*, or* S4-1 *in* $O(m + n)$ *time.*

Proof. Given G in Case S4-2 as input, the reduction algorithm is as follows:

```
Step 1. Construct $\Delta(G)$.
Step 2. repeat
        (a) Use Lemma 4.3(1) to reroot $\Delta(G)$.
        (b) Use Lemma 4.3(2) to find a legal pair $w_1$ and $w_2$.
        (c) Add a binding edge $e$ for $w_1$ and $w_2$ into $G$.
        (d) Use Lemma 2.2 to update $\Delta(G)$ while rerooting it at the new b-vertex
            $Y_e$ resulting from the insertion of $e$.
        until $G$ does not satisfy Case S4-2.
```

Since Step 1 takes $O(m+n)$ time, it suffices to prove that Step 2 takes $O(n)$ time. By Lemma 3.9(1), each iteration of Step 2 reduces $|\Lambda(G)|$ by two. Since $|\Lambda(G)| < n$, the repeat loop has less than n iterations. Then, since the until condition can be checked in $O(1)$ time per iteration using Lemma 2.4(2) and items 2 and 3 of $\Delta(G)$, the until step takes $O(n)$ total time. Similarly, Step 2(c) takes $O(1)$ time per iteration and $O(n)$ total time in a straightforward manner.

We next show that Steps 2(a), 2(b) and 2(d) also take $O(n)$ total time. For a given iteration, let G_0 and G_1 denote G before and after e is inserted, respectively.

Step 2(a). We show that each case in the proof of Lemma 4.3(1) takes $O(n)$ total time as follows.

Case 1: This case takes $O(1)$ time per iteration and thus $O(n)$ total time.

Case 2: By Lemma 3.10(3), this case can only happen once in the above reduction algorithm. Hence, this case takes $O(n)$ total time.

Case 3: This case takes $O(1)$ time per edge on $P_{r,h}$ for an iteration. Note that the degree of a vertex in $\Delta(G)$ never increases by edge insertion. Then, since $\Delta(G_1)$ is rooted at Y_e with e connecting two leaves of $\Delta(G_0)$, each edge on $P_{r,h}$ is traversed only once to reroot $\Delta(G)$ for this case throughout all the iterations. Therefore, this case takes $O(n)$ total time.

Step 2(b). This step takes $O(|P_{w_1,w_2}|)$ time per iteration. Since there are $O(n)$ iterations, by Lemma 2.2(4), this step takes $O(n)$ total time.

Step 2(d). We bound the time for updating each item of $\Delta(G)$ as follows.

Item 1 of $\Delta(G)$. Notice that P_{w_1,w_2} passes through the root of $\Delta(G_0)$. Also, $\Delta(G_1)$ is rooted at Y_2. These properties make it straightforward to update this item in $O(|P_{w_1,w_2}|)$ time per iteration. Since there are $O(n)$ iterations, by Lemma 2.2(4), this step takes $O(n)$ total time.

Item 2 of $\Delta(G)$. By Lemma 3.9(1), $\Lambda(G_1) = \Lambda(G_0) - \{w_1, w_2\}$. Thus it takes $O(1)$ time to update this item per iteration and $O(n)$ total time.

Item 3 of $\Delta(G)$. Let u be a c-vertex in $\Delta(G_0)$. If $u \notin P_{w_1,w_2}$, it has the same degree in $\Delta(G_0)$ and $\Delta(G_1)$ and is not relocated in this item. If $u \in P_{w_1,w_2}$, its degree reduces at most two in $\Delta(G_1)$ and can be relocated in $O(1)$ time. Therefore, this item can be updated in $O(|P_{w_1,w_2}|)$ time per iteration, i.e., $O(n)$ total time as shown for item 1. □

Acknowledgments. We are very grateful to Dan Gusfield for insightful discussions and to the anonymous referee for extremely thorough comments.

REFERENCES

[1] N. R. Adam and J. C. Wortmann, *Security-control methods for statistical database: A comparative study*, ACM Computing Surveys, 21 (1989), pp. 515–556.

[2] F. Y. Chin and G. Özsoyoğlu, *Auditing and inference control in statistical databases*, IEEE Transactions on Software Engineering, 8 (1982), pp. 574–582.

[3] L. H. Cox, *Suppression methodology and statistical disclosure control*, J. Amer. Statist. Assoc., 75 (1980), pp. 377–385.

[4] D. E. Denning and J. Schlörer, *Inference controls for statistical databases*, IEEE Trans. Computer, 16 (1983), pp. 69–82.

[5] K. P. Eswaran and R. E. Tarjan, *Augmentation problems*, SIAM J. Comput., 5 (1976), pp. 653–665.

[6] A. Frank, *Augmenting graphs to meet edge-connectivity requirements*, SIAM J. Discrete Math., 5 (1992), pp. 25–53.

[7] A. Frank, *Connectivity augmentation problems in network design*, in Mathematical Programming: State of the Art 1994, J. R. Birge and K. G. Murty, eds., University of Michigan, Ann Arbor, MI, 1994, pp. 34–63.

[8] D. Gusfield, *A graph theoretic approach to statistical data security*, SIAM J. Comput., 17 (1988), pp. 552–571.

[9] F. Harary, *Graph Theory*, Addison-Wesley, Reading, MA, 1969.

[10] T.-S. Hsu, *Graph Augmentation and Related Problems: Theory and Practice*, Ph.D. thesis, University of Texas at Austin, 1993.

[11] T.-S. Hsu, *Undirected vertex-connectivity structure and smallest four-vertex-connectivity augmentation (extended abstract)*, in Proceedings of the 6th Annual International Symposium on Algorithms and Computation, Lecture Notes in Comput. Sci. 1004, J. Staples, P. Eades, N. Katoh, and A. Moffat, eds., Springer-Verlag, New York, 1995, pp. 274–283.

[12] T.-S. Hsu, *On four-connecting a triconnected graph*, J. Algorithms, 35 (2000), pp. 202–234.

[13] T.-S. Hsu, *Simpler and faster vertex-connectivity augmentation algorithms (extended abstract)*, in Proceedings of the 8th Annual European Symposium on Algorithms, Lecture Notes in Comput. Sci. 1879, M. Paterson, ed., Springer-Verlag, New York, 2000, pp. 278–289.

[14] T.-S. Hsu, *Simpler and faster biconnectivity augmentation*, J. Algorithms, 45 (2002), pp. 55–71.

[15] T.-S. Hsu and M. Y. Kao, *Optimal augmentation for bipartite componentwise biconnectivity in linear time*, in Proceedings of the 7th Annual International Symposium on Algorithms and Computation, Lecture Notes in Comput. Sci. 1178, T. Asano, Y. Igarashi, H. Nagamochi, S. Miyano, and S. Suri, eds., Springer-Verlag, New York, 1996, pp. 213–222.

[16] T.-S. Hsu and M. Y. Kao, *Optimal bi-level augmentation for selectively enhancing graph connectivity with applications*, in Proceedings of the 2nd Annual International Computing and Combinatorics Conference, Lecture Notes in Comput. Sci. 1090, J. Y. Cai and C. K. Wong, eds., Springer-Verlag, New York, 1996, pp. 169–178.

[17] T.-S. Hsu and M. Y. Kao, *Security problems for statistical databases with general cell suppressions*, in Proceedings of the 9th International Conference on Scientific and Statistical Database Management, D. Hansen and Y. Ioannidis, eds., IEEE Computer Society Press, Los Alamitos, CA, 1997, pp. 155–164.

[18] T.-S. Hsu and M. Y. Kao, *A unifying augmentation algorithm for two-edge connectivity and biconnectivity*, J. Comb. Optim., 2 (1998), pp. 237–256.

[19] T.-S. Hsu and V. Ramachandran, *A linear time algorithm for triconnectivity augmentation*, in Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1991, pp. 548–559.

[20] T.-S. Hsu and V. Ramachandran, *Finding a smallest augmentation to biconnect a graph*, SIAM J. Comput., 22 (1993), pp. 889–912.

[21] G. Kant, *Algorithms for Drawing Planar Graphs*, Ph.D. thesis, Utrecht University, Utrecht, the Netherlands, 1993.

[22] M. Y. Kao, *Linear-time optimal augmentation for componentwise bipartite-completeness of graphs*, Inform. Process. Lett., (1995), pp. 59–63.

[23] M. Y. Kao, *Data security equals graph connectivity*, SIAM J. Discrete Math., 9 (1996), pp. 87–100.

[24] M. Y. Kao, *Total protection of analytic-invariant information in cross-tabulated tables*, SIAM J. Comput., 26 (1997), pp. 231–242.

[25] J. P. Kelly, B. L. Golden, and A. A. Assad, *Cell suppression: Disclosure protection for sensitive tabular data*, Networks, 22 (1992), pp. 397–417.

[26] F. M. Malvestuto and M. Moscarini, *Censoring statistical tables to protect sensitive information: Easy and hard problems*, in Proceedings of the 8th International Conference on Scientific and Statistical Database Management, IEEE Computer Society Press, Los Alamitos, CA, 1996, pp. 12–21.

[27] F. M. Malvestuto and M. Moscarini, *Suppressing marginal totals from a two-dimensional table to protect sensitive information*, Stat. Comput., 7 (1997), pp. 101–114.

[28] F. M. Malvestuto, M. Moscarini, and M. Rafanelli, *Suppressing marginal cells to protect sensitive information in a two-dimensional statistical table*, in Proceedings of the 10th ACM-SIGMOD-SIGACT Symposium on Principles of Database Systems, Denver, CO, 1991, pp. 252–258.

[29] A. Rosenthal and A. Goldner, *Smallest augmentations to biconnect a graph*, SIAM J. Comput., 6 (1977), pp. 55–66.

[30] S. Taoka and T. Watanabe, *Minimum augmentation to k-edge-connect specified vertices of a graph*, in Proceedings of the 5th Annual International Symposium on Algorithms and Computation, Lecture Notes in Comput. Sci. 834, D. Z. Du and X. S. Zhang, eds., Springer-Verlag, New York, 1994, pp. 217–225.

[31] T. Watanabe, Y. Higashi, and A. Nakamura, *An approach to robust network construction from graph augmentation problems*, in Proceedings of the 1990 IEEE International Symposium on Circuits and Systems, IEEE Computer Society Press, Los Alamitos, CA, 1990, pp. 2861–2864.

[32] T. Watanabe, Y. Higashi, and A. Nakamura, *Graph augmentation problems for a specified set of vertices*, in Proceedings of the 1st Annual International Symposium on Algorithms, Lecture Notes in Comput. Sci. 450, T. Asano, T. Ibaraki, H. Imai, and T. Nishizeki, eds., Springer-Verlag, New York, 1990, pp. 378–387.

[33] T. Watanabe and A. Nakamura, *A minimum* 3*-connectivity augmentation of a graph*, J. Comput. System Sci., 46 (1993), pp. 91–128.

[34] T. Watanabe, S. Taoka, and T. Mashima, *Minimum-cost augmentation to* 3*-edge-connect all specified vertices in a graph*, in Proceedings of the 1993 IEEE International Symposium on Circuits and Systems, IEEE Computer Society Press, Los Alamitos, CA, 1993, pp. 2311–2314.

SIAM J. Discrete Math.
Vol. 19, No. 2, pp. 363–370

CLASSIFICATION OF SELF-ORTHOGONAL CODES OVER $\mathbb{F}_3$ AND $\mathbb{F}_4$*

ILIYA BOUYUKLIEV† AND PATRIC R. J. ÖSTERGÅRD‡

Abstract. Several methods for classifying self-orthogonal codes up to equivalence are presented. These methods are used to classify self-orthogonal codes with largest possible minimum distance over the fields $\mathbb{F}_3$ and $\mathbb{F}_4$ for lengths $n \leq 29$ and small dimensions (up to 6). Some properties of the classified codes are also presented. In particular, an extensive collection of quantum error-correcting codes is obtained.

Key words. code equivalence, quantum codes, self-dual codes, self-orthogonal codes

AMS subject classifications. 94B05, 94B65, 81P68

DOI. 10.1137/S0895480104441085

1. Introduction. Let $\mathbb{F}_q^n$ denote the vector space of n-tuples over the q-element field $\mathbb{F}_q$. A q-ary linear code C of length n and dimension k, or an $[n,k]_q$ code, is a k-dimensional subspace of $\mathbb{F}_q^n$. An inner product $(\mathbf{x},\mathbf{y})$ of vectors $\mathbf{x},\mathbf{y} \in \mathbb{F}_q^n$ defines orthogonality: Two vectors are said to be orthogonal if their inner product is 0. The set of all vectors of $\mathbb{F}_q^n$ orthogonal to all codewords from C is called the orthogonal code $C^\perp$ to C:

$$C^\perp = \{\mathbf{x} \in \mathbb{F}_q^n \mid (\mathbf{x},\mathbf{y}) = 0 \text{ for any } \mathbf{y} \in C\}.$$

It is well known that the code $C^\perp$ is a linear $[n, n-k]_q$ code.

A $k \times n$ matrix $\mathbf{G}_C$ whose rows form a basis of C is called a generator matrix of C. A generator matrix of the code $C^\perp$, orthogonal to C, is a parity check matrix for C, denoted by $\mathbf{H}_C$.

The number of nonzero coordinates of a vector $\mathbf{x} \in \mathbb{F}_q^n$ is called its Hamming weight $\mathrm{wt}(\mathbf{x})$. The Hamming distance $d(\mathbf{x},\mathbf{y})$ between two vectors $\mathbf{x},\mathbf{y} \in \mathbb{F}_q^n$ is defined by

$$d(\mathbf{x},\mathbf{y}) = \mathrm{wt}(\mathbf{x}-\mathbf{y}).$$

The minimum distance of a linear code C is

$$d(C) = \min\{d(\mathbf{x},\mathbf{y}) \mid \mathbf{x},\mathbf{y} \in C, \mathbf{x} \neq \mathbf{y}\} = \min\{\mathrm{wt}(\mathbf{c}) \mid \mathbf{c} \in C, \mathbf{c} \neq \mathbf{0}\}.$$

A q-ary linear code of length n, dimension k, and minimum distance d is said to be an $[n,k,d]_q$ code.

If $C \subseteq C^\perp$, then the code C is called *self-orthogonal*. Self-orthogonal codes with $n = 2k$ are of particular interest; then $C = C^\perp$ and the codes are called *self-dual*. The

*Received by the editors February 13, 2004; accepted for publication (in revised form) December 21, 2004; published electronically October 20, 2005.
http://www.siam.org/journals/sidma/19-2/44108.html

†Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, P.O. Box 323, 5000 V. Tarnovo, Bulgaria (iliya@moi.math.bas.bg). The work of this author was partially supported by Bulgarian National Science Fund contract MM1304/2003.

‡Department of Electrical and Communications Engineering, Helsinki University of Technology, P.O. Box 3000, 02015 HUT, Finland (patric.ostergard@hut.fi). The work of this author was partially supported by Academy of Finland grants 100500 and 202315.

classes of self-orthogonal and self-dual codes are important in coding theory from both a practical and a theoretical point of view. Self-dual codes over $\mathbb{F}_3$ are particularly interesting because they include the Golay code of length 12, quadratic residue codes, and symmetry codes. Many papers have been devoted to the study of self-dual codes; see the excellent survey [18] for an overview of these results.

In this work ternary and quaternary self-orthogonal codes with maximum possible minimum distance are considered. Such codes are classified for lengths $n \leq 29$ and dimensions $k \leq 6$ (except for sets of parameters where this was not computationally feasible using our algorithms). Two different methods are used in the classification, so (in nearly all cases) the results have been obtained by two independent algorithms. Very little has been known about the minimum distance and number of self-orthogonal codes, since most of the research has been into the special case of self-dual codes. For general linear codes, extensive tables of bounds can be found in [3].

In section 2, two types of inner products are defined and some properties of the weight distributions of self-orthogonal codes are presented. Two classification methods are considered in section 3, and the computational results obtained by these methods are tabulated in section 4. Finally, in section 5 some data for quantum error-correcting codes obtained from the classified quaternary self-orthogonal codes are presented.

2. Preliminaries. The *Euclidean inner product* of two vectors $\mathbf{u} = (u_1, u_2, \ldots, u_n)$ and $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ from $\mathbb{F}_q^n$ is defined by

$$(\mathbf{u}, \mathbf{v})_E = u_1 v_1 + u_2 v_2 + \cdots + u_n v_n.$$

For codes over $\mathbb{F}_q$ where q is an even power of an arbitrary prime p, one can consider another type of inner product, the Hermitian inner product. The *Hermitian inner product* of two vectors $\mathbf{u} = (u_1, u_2, \ldots, u_n)$ and $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ from $\mathbb{F}_q^n$ is defined by

$$(\mathbf{u}, \mathbf{v})_H = u_1 \bar{v}_1 + u_2 \bar{v}_2 + \cdots + u_n \bar{v}_n,$$

where $\bar{v}_i = v_i^{\sqrt{q}}$ for $v_i \in \mathbb{F}_q$. Consequently, for $q = 4$ the Hermitian inner product is defined by

$$(\mathbf{u}, \mathbf{v})_H = u_1 v_1^2 + u_2 v_2^2 + \cdots + u_n v_n^2.$$

In the ternary case we consider the Euclidean inner product, and in the quaternary case (like in most other studies) we consider the Hermitian inner product. Throughout the paper, these inner products are assumed in the discussion of self-dual and self-orthogonal codes.

The MacWilliams identities can give a lot of information about possible weight distributions of self-dual codes. However, in the current study we do not use this information. We shall now present some known basic results on the codes considered here. For a proof of Lemma 1, see [8, Theorem 1.4.10]. A code is even (resp., doubly-even) if the weights of all codewords are divisible by 2 (resp., 4).

Lemma 1. *Let C be a code over $\mathbb{F}_q$ with $q = 3$ or 4.*

(i) *When $q = 3$, every codeword of C has weight divisible by three if and only if C is self-orthogonal.*

(ii) *When $q = 4$, every codeword of C has weight divisible by two if and only if C is Hermitian self-orthogonal.*

3. Two classification methods. In any work on classifying mathematical objects, one should carefully define the concept of equivalence (or, depending on the conventional terminology, isomorphism). For self-dual and self-orthogonal codes, the definition of equivalence depends on the inner product. For ternary codes with Euclidean inner product and quaternary codes with Hermitian inner product, the definition coincides with the definition for general linear codes [18].

Two linear q-ary codes, C_1 and C_2, are said to be *equivalent* if the codewords of C_2 can be obtained from the codewords of C_1 via a sequence of transformations of the following types:

1. permutation of coordinates,
2. multiplication of the elements in a given coordinate by a nonzero element of $\mathbb{F}_q$,
3. application of a field automorphism to the elements in all coordinates simultaneously.

(The field $\mathbb{F}_3$ does not have nontrivial automorphisms, and the only nontrivial automorphism of $\mathbb{F}_4$ is conjugation.) An *automorphism* of a linear code C is a sequence of such transformations that maps each codeword of C onto a codeword of C. The automorphisms of a code C form a group, called the automorphism group of the code and denoted by $\mathrm{Aut}(C)$.

Determining equivalence of codes plays a central role in any classification algorithm. Not only must one make sure that all completed codes are inequivalent, but determining equivalence of partial codes is also important for efficiency reasons. The first author used an algorithm for determining code equivalence that was developed in [1] and is based on the ideas in [12]. The approach of the second author depends on the graph isomorphism program *nauty* [12, 13] for this matter; see [15] for further details (but some enhancements of the basic method will be presented here).

The approaches to be presented differ in the ways the codes are built up via smaller codes. For efficiency reasons, Lemma 1 should be taken into account.

The first approach uses results on the parameters of residuals of codes. Let $\mathbf{G}$ be a generator matrix of a linear $[n,k,d]_q$ code C. Then the *residual code* $\mathrm{Res}(C,\mathbf{c})$ of C with respect to a codeword $\mathbf{c}$ is the code generated by the restriction of $\mathbf{G}$ to the columns where $\mathbf{c}$ has a zero entry. The following result is from [6].

LEMMA 2. *Suppose C is an $[n,k,d]_q$ code and suppose $\mathbf{c} \in C$ has weight w, where $d > w(q-1)/q$. Then Res$(C,\mathbf{c})$ is an $[n-w,k-1,d']_q$ code with $d' \geq d-w+\lceil w/q \rceil$.*

In addition to constructing $[n,k,d]_q$ codes from their $[n-w,k-1,d']_q$ residual codes, one may also start from $[n-i,k,d']_q$ codes. On the bottom of this hierarchy of extensions is the trivial $[k,k,1]_q$ code.

In the second approach, $[n,k,d]_q$ codes are constructed by extending $[n{-}i,k{-}i,d]_q$ or $[n-i-1,k-i,d]_q$ codes. The following result shows when the latter type of code can be used [10, p. 592].

LEMMA 3. *Let C be an $[n,k,d]_q$ code. If there exists a codeword $\mathbf{c} \in C^{\perp}$ with $wt(\mathbf{c}) = i$, then there is an $[n-i,k-i+1,d]_q$ code.*

If $\mathbf{G}$ is a generator matrix for an $[n-i,k-i,d]_q$ or an $[n-i-1,k-i,d]_q$ code, we extend it (in all possible ways) to

$$(1) \qquad \left(\begin{array}{c|c} * & \mathbf{I}_i \\ \hline \mathbf{G} & \mathbf{0} \end{array}\right) \quad \text{or} \quad \left(\begin{array}{c|c} * & \mathbf{1}\ \mathbf{I}_i \\ \hline \mathbf{G} & \mathbf{0} \end{array}\right),$$

respectively, where $\mathbf{I}_i$ is the $i \times i$ identity matrix, $\mathbf{1}$ is an all-1 column vector, and the starred submatrix is to be determined. If we let the matrix $\mathbf{G}$ be in systematic form, we can fix k more columns to get

$$\left(\begin{array}{c|c|c} * & \mathbf{0} & \mathbf{I}_i \\ \hline \mathbf{G}_1 & \mathbf{I}_k & \mathbf{0} \end{array}\right) \quad \text{or} \quad \left(\begin{array}{c|c|c} * & \mathbf{0} & \mathbf{1}\,\mathbf{I}_i \\ \hline \mathbf{G}_1 & \mathbf{I}_k & \mathbf{0} \end{array}\right). \tag{2}$$

More information on this approach can be found in [2]. The subcodes through which the codes are constructed must also be self-orthogonal. For the approach via residual codes, on the other hand, such a restriction does not apply.

If $i = 1$ in the second approach, we get the method used in [15], where $[n, k, d]_q$ codes are obtained from $[n-1, k-1, d]_q$ codes by adding a new column in all possible ways to the parity check matrix, checking the minimum distance and orthogonality of the new code, and finally removing copies of equivalent codes. We shall now see how the equivalence test can be enhanced for this particular variant.

To speed up the algorithm and reduce the need for extensive tables of intermediate codes, a classification technique developed by McKay [14] was implemented. Essentially, the idea is that (in our case) a code can be obtained from several subcodes, only one of which is identified as the "parent" of the new code. Then a new code is rejected unless it was obtained from its parent. Note that identifying a certain subcode essentially means identifying a coordinate, and with the encoding used in [15] the output of *nauty* can be used to get a canonical labelling of the coordinates.

Shortening an $[n, k, d]_q$ linear code by deleting one coordinate and keeping the codewords with a 0 in the given coordinate gives a $[n-1, k', d]_q$ code with $k' = k$ if the original code has only zeroes in the coordinate to be deleted and $k' = k-1$ otherwise. Therefore, in the parent test of a McKay-type algorithm—after adding one coordinate via a new column in the parity check matrix—one should first check which coordinates are all-zero. In the test itself, only coordinates that are not all-zero should be considered. For fields with nontrivial automorphisms, like $\mathbb{F}_4$, if one uses the idea of producing one graph for each automorphism [15], a code passes the parent test if at least one of the $|\mathrm{Aut}(\mathbb{F}_q)|$ instances passes the test.

4. Results.

We first give a short overview of old results on classifying ternary and quaternary self-dual and self-orthogonal codes. See [18] for more details and references.

The length of any ternary self-dual code is divisible by 4, and this necessary condition is also sufficient. Such codes of length less than or equal to 20—and self-orthogonal codes of maximal dimension and length less than or equal to 19—have been completely classified in [5, 11, 16, 17]. A partial classification of the self-dual codes of length 24 can be found in [9], including a classification of such codes with maximum minimum distance. For ternary self-dual codes, $d \leq 3\left\lfloor \frac{n}{12} \right\rfloor + 3$ holds [18, Theorem 28]. Codes meeting this bound are called *extremal* and are known to exist for admissible lengths $n \leq 48$, $56 \leq n \leq 64$ and do not exist for $n = 72, 96, 120$ and $n \geq 144$.

Quaternary self-dual codes have even lengths. They have been classified up to length 16 in [5]. For quaternary self-dual codes, $d \leq 2\left\lfloor \frac{n}{6} \right\rfloor + 2$ holds [18, Theorem 28]. Extremal codes (which meet this bound) exist for admissible lengths $n \leq 10$, $14 \leq n \leq 22$, and $n = 28, 30$ and do not exist for $n = 12, 24, 102, 108, 114, 120, 122$ and $n \geq 126$.

There are only sporadic classification results for ternary and quaternary self-orthogonal codes in the literature [7]. This work makes a contribution toward filling this gap.

The classification results are presented in Tables 1 and 2. For lengths $n \leq 29$ and dimensions $3 \leq k \leq 6$, the maximal minimum distance and the number of corresponding codes are shown. Entries that could not be computed with a reasonable amount of CPU time are empty. For such instances, one could consider a subclass of codes. We considered *doubly even* self-orthogonal quaternary codes to find out that the number of $[20, 4, 12]_4$, $[20, 5, 12]_4$, $[21, 5, 12]_4$, $[25, 4, 16]_4$, and $[25, 5, 16]_4$ such codes is 16, 4, 4, 333, and 31, respectively.

5. Quaternary self-orthogonal codes and quantum codes. Quantum computers have received a lot of attention in the last decade, after Shor proved that integer factorization can be solved in polynomial time on such computers [19]. The quantum analogue of a bit of information is called a *qubit* and is the state of a system in a two-dimensional Hilbert space spanned by e_0 and e_1, where e_0 and e_1 are eigenvectors corresponding to the eigenvalues 0 and 1 of the qubit.

The setting in which quantum error-correcting codes (QECCs) exist is the quantum state space of n qubits (quantum bits, or 2-state quantum systems). This space is $\mathbb{C}^{2^n}$, and it has a natural decomposition as the tensor product of n copies of $\mathbb{C}^2$, where each copy corresponds to one qubit. Many known quantum codes have close connections to a finite group of unitary transformations of $\mathbb{C}^{2^n}$, known as a Clifford group.

A QECC is defined to be a unitary mapping (encoding) of k qubits (2-state quantum systems) into a subspace of the quantum state space of n qubits such that if any t of the qubits undergo arbitrary decoherence, not necessarily independently, the resulting n qubits can be used to faithfully reconstruct the original quantum state of the k encoded qubits. In general, by $[[n, k, d]]$ we denote a QECC that encodes k qubits of a quantum system into n qubits. The parameter d is the minimum distance of the code. A QECC with minimum distance d can be used to detect errors that involve at most $d - 1$ of the n subsystems. Alternatively, one can correct errors that involve at most $\lfloor (d - 1)/2 \rfloor$ subsystems. See [4] for more information about QECCs.

It is known that if C is a Hermitian self-orthogonal linear $[n, k]_4$ code such that there are no vectors of weight $< d$ in $C^\perp \setminus C$ (where $C^\perp$ is the Hermitian dual of C), then there is a quantum error-correcting $[[n, n - 2k, d]]$ code [4]. By investigating the classified codes with respect to this property, a number of quantum error-correcting codes were detected. The parameters of these codes with $d \geq 3$ are given in Table 3.

The first column of Table 3 shows the parameters of the quaternary codes, and the parameters of the corresponding quantum codes are given in the second column. The orders of the automorphism groups of the quaternary codes with dual distance at least 3 are given in the third column—an upper index gives the number of corresponding codes—and the last column lists the number of quaternary codes with maximum possible dual distance.

Table 1
Classification of ternary self-orthogonal codes.

$n\backslash k$	3		4		5		6	
7	3	1						
8	3	1	3	1				
9	6	1	3	1				
10	6	1	6	1				
11	6	2	6	1	6	1		
12	6	6	6	6	6	1	6	1
13	9	1	6	10	6	4	6	1
14	9	1	6	27	6	15	6	4
15	9	4	9	3	6	73	6	20
16	9	9	9	13	9	1	6	121
17	9	16	9	58	9	35	6	885
18	12	2	9	308	9	997	9	105
19	12	4	12	1	9	15207	9	18019
20	12	14	12	32	12	2	9	
21	12	36	12	406	12	359	9	
22	15	1	12	3679	12	107017	12	698
23	15	3	12	20673	12		12	
24	15	15	15	13	12		12	
25	18	45	15	699	15	23	12	
26	18	1	15	17703	15		15	2
27	18	4	18	1	15		15	
28	18	14	18	6	15		15	
29	18	49	18	406	18	1	15	

Table 2
Classification of quaternary (Hermitian) self-orthogonal codes.

$n\backslash k$	3		4		5		6	
6	4	1						
7	4	1						
8	4	4	4	1				
9	6	1	4	2				
10	6	4	4	12	4	2		
11	6	6	6	2	4	6		
12	8	5	6	22	6	2	4	5
13	8	10	8	5	6	19	6	1
14	10	1	8	92	8	4	6	23
15	10	7	8	911	8	460	8	3
16	12	1	10	50	8	45311	8	1081
17	12	4	12	1	10	91	8	
18	12	45	12	12	10		10	3
19	12	185	12	5673	10		10	
20	14	10	12		12		10	
21	16	1	14	212	12		12	
22	16	4	14		14	67	12	
23	16	46	16	3	14		12	
24	16	614	16	40397	16		14	
25	18	6	16		16		14	
26	18	185	18	14	16		16	
27	20	2	18		16		16	
28	20	46	20	1	18		16	
29	20	850	20	22656	18		16	

Table 3
Quaternary self-orthogonal codes and quantum codes.

C	D	$\lvert \mathrm{Aut}(C) \rvert$	#
$[6,3,4]$	$[[6,0,3]]$	2160	1
$[7,3,4]$	$[[7,1,3]]$	1008	1
$[8,3,4]$	$[[8,2,3]]$	1728	1
$[8,4,4]$	$[[8,0,4]]$	8064	1
$[9,3,6]$	$[[9,3,3]]$	1296	1
$[9,4,4]$	$[[9,1,3]]$	$4320, 1152$	2
$[10,3,6]$	$[[10,4,3]]$	360	1
$[10,4,4]$	$[[10,2,3]]$	$1728, 192, 432, 259200$	4
$[10,5,4]$	$[[10,0,4]]$	$43200, 11520$	2
$[11,3,6]$	$[[11,5,3]]$	360	1
$[11,4,6]$	$[[11,3,3]]$	36	1
$[11,5,4]$	$[[11,1,3]]$	$1728, 12096, 8640, 576, 11520, 777600$	6
$[12,3,8]$	$[[12,6,3]]$	1296	1
$[12,4,6]$	$[[12,4,4]]$	$18, 144^2, 576, 12^2, 24, 1296, 720$	1
$[12,5,6]$	$[[12,2,4]]$	$72, 216$	1
$[12,6,4]$	$[[12,0,4]]$	$20736, 60480, 6912, 138240, 9331200$	5
$[13,3,8]$	$[[13,7,3]]$	1728	1
$[13,4,8]$	$[[13,5,3]]$	$24^2, 36, 432, 720$	5
$[13,5,6]$	$[[13,3,4]]$	$72, 12, 36^3, 144, 6^5, 24, 18^3, 48$	1
$[13,6,6]$	$[[13,1,5]]$	468	1
$[14,3,10]$	$[[14,8,3]]$	1008	1
$[14,4,8]$	$[[14,6,4]]$	$3^5, 12^8, 48^5, 24^3, 6^8, 432^2,$ $42, 9, 36, 144^3, 18, 192, 3456, 8064$	1
$[14,5,8]$	$[[14,4,4]]$	$36, 24, 72, 288$	4
$[14,6,6]$	$[[14,2,5]]$	$6^2, 48, 36^8, 12^3, 18, 72, 24, 144^2, 252$	1
$[15,3,10]$	$[[15,9,3]]$	2160	1
$[15,4,8]$	$[[15,7,3]]$	$24^5, 6^{52}, 12^{30}, 48^6, 2160, 3^{74}, 18^6, 36^4, 9^2$ $144^3, 192, 1152, 60, 432, 504, 120960$	189
$[15,5,8]$	$[[15,5,4]]$	$24^7, 48, 6^{95}, 3^{258}, 12^{22}, 9^5, 18^5$ $108, 72^3, 216, 30^2, 60, 36, 360, 288, 15$	26
$[15,6,8]$	$[[15,3,5]]$	$216, 72, 360$	3
$[16,3,12]$	$[[16,10,3]]$	17280	1
$[16,4,10]$	$[[16,8,3]]$	$3^{18}, 6^{10}, 24^2, 18, 96, 12^3, 9, 36, 72$	38
$[16,5,8]$	$[[16,6,4]]$	$48^{32}, 24^{118}, 6^{2824}, 3^{27856}, 12^{496}, 72^9$ $36^{19}, 9^{31}, 576^2, 192^6, 96^{18}, 18^{24}, 60^3, 216^3, 360$ $1080^2, 30^3, 288^2, 144^3, 15, 1728, 2160, 768^3, 3072$ $384^2, 18432, 2304^3, 54, 1152, 8064, 1935360$	519
$[16,6,8]$	$[[16,4,4]]$	$12^{52}, 48^2, 96^6, 768, 36^{12}, 864, 3^{686}, 6^{259}, 144^2$ $24^{14}, 108, 9^3, 18^{15}, 72^2, 288, 384, 4608$	697
$[17,4,12]$	$[[17,9,4]]$	48960	1
$[17,5,10]$	$[[17,7,4]]$	$3^{82}, 6^6, 9^2, 126$	27
$[18,4,12]$	$[[18,10,3]]$	$24, 12, 72^3, 36^2, 60, 360, 432, 6480$	11
$[18,6,10]$	$[[18,6,5]]$	$18, 54, 108$	2
$[19,4,12]$	$[[19,11,3]]$	$3^{2111}, 18^9, 6^{350}, 36^{11}, 12^{55}, 72^7, 9^{10}, 144, 24^{12}, 27, 48^3$	2570
$[21,3,16]$	$[[21,15,3]]$	362880	1
$[21,4,14]$	$[[21,13,3]]$	$12^6, 6^{28}, 3^{169}, 9^4, 24, 42, 18, 63, 60$	212
$[22,5,14]$	$[[22,12,4]]$	$3^{42}, 6^{18}, 18^4, 9, 12, 36$	67
$[23,4,16]$	$[[23,15,3]]$	24	1
$[24,4,16]$	$[[24,16,3]]$	$6^{1302}, 60, 3^{18934}, 12^{139}, 18^{13}, 24^{28}, 9^{12}, 36^5$ $72^4, 288, 120, 192^3, 48^4, 1152, 144^2, 90$ $162, 414720, 576, 96, 648$	20456
$[26,4,18]$	$[[26,18,3]]$	$3^9, 12, 6^2, 18, 24$	14
$[28,4,20]$	$[[28,20,3]]$	42	1
$[29,4,20]$	$[[29,21,3]]$	$6^{840}, 3^{10385}, 12^{111}, 24^{10}, 18^5, 9^4, 36^5, 48^3, 72, 21$	11365

REFERENCES

[1] I. Bouyukliev, *An algorithm for finding isomorphisms of codes*, in Proceedings of the International Workshop OCRT, Sunny Beach, Bulgaria, 2001, pp. 35–41.

[2] I. Bouyukliev and J. Simonis, *Some new results for optimal ternary linear codes*, IEEE Trans. Inform. Theory, 48 (2002), pp. 981–985.

[3] A. E. Brouwer, *Bounds on the size of linear codes*, in Handbook of Coding Theory, V. S. Pless and W. C. Huffman, eds., Elsevier, Amsterdam, 1998, pp. 295–461.

[4] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, *Quantum error correction via codes over GF*(4), IEEE Trans. Inform. Theory, 44 (1998), pp. 1369–1387.

[5] J. H. Conway, V. Pless, and N. J. A. Sloane, *Self-dual codes over GF*(3) *and GF*(4) *of length not exceeding* 16, IEEE Trans. Inform. Theory, 25 (1979), pp. 312–322.

[6] S. Dodunekov, *Minimal block length of a linear q-ary code with specified dimension and code distance*, Probl. Inf. Transm., 20 (1984), pp. 239–249.

[7] M. van Eupen and P. Lisoněk, *Classification of some optimal ternary linear codes of small length*, Des. Codes Cryptogr., 10 (1997), pp. 63–84.

[8] W. C. Huffman and V. Pless, *Fundamentals of Error-Correcting Codes*, Cambridge University Press, Cambridge, UK, 2003.

[9] J. S. Leon, V. Pless, and N. J. A. Sloane, *On ternary self-dual codes of length* 24, IEEE Trans. Inform. Theory, 27 (1981), pp. 176–180.

[10] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1977.

[11] C. L. Mallows, V. Pless, and N. J. A. Sloane, *Self-dual codes over GF*(3), SIAM J. Appl. Math., 31 (1976), pp. 649–666.

[12] B. D. McKay, *Practical graph isomorphism*, Congr. Numer., 30 (1981), pp. 45–87.

[13] B. D. McKay, *nauty* User's Guide (version 1.5), Tech. report TR-CS-90-02, Computer Science Department, Australian National University, Canberra, Australia, 1990.

[14] B. D. McKay, *Isomorph-free exhaustive generation*, J. Algorithms, 26 (1998), pp. 306–324.

[15] P. R. J. Östergård, *Classifying subspaces of Hamming spaces*, Des. Codes Cryptogr., 27 (2002), pp. 297–305.

[16] V. Pless, *On the uniqueness of the Golay codes*, J. Combin. Theory, 5 (1968), pp. 215–228.

[17] V. Pless, N. J. A. Sloane, and H. N. Ward, *Ternary codes of minimum weight* 6 *and the classification of self-dual codes of length* 20, IEEE Trans. Inform. Theory, 26 (1980), pp. 305–316.

[18] E. M. Rains and N. J. A. Sloane, *Self-dual codes*, in Handbook of Coding Theory, V. S. Pless and W. C. Huffman, eds., Elsevier, Amsterdam, 1998, pp. 177–294.

[19] P. W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. Comput., 26 (1997), pp. 1484–1509.

SIAM J. Discrete Math.
Vol. 19, No. 2, pp. 371–381

ON UNAVOIDABLE SETS OF WORD PATTERNS*

ALEXANDER BURSTEIN[†] AND SERGEY KITAEV[‡]

Abstract. We introduce the notion of unavoidable (complete) sets of word patterns, which is a refinement for that of words, and study certain numerical characteristics for unavoidable sets of patterns. In some cases we employ the graph of pattern overlaps introduced in this paper, which is a subgraph of the de Bruijn graph and which we prove to be Hamiltonian. In other cases we reduce a problem under consideration to known facts on unavoidable sets of words. We also give a relation between our problem and the extensively studied universal cycles and prove that there exists a universal cycle for word patterns of any length over any alphabet. The Stirling numbers of the second kind and the Möbius function appear in our results.

Key words. pattern, word, (un)avoidability, de Bruijn graph, universal cycles, Stirling numbers of the second kind

AMS subject classifications. 05A05, 05A15, 68R15

DOI. 10.1137/S0895480104445678

1. Introduction.

When defining or characterizing sets of objects in discrete mathematics, "languages of prohibitions" are often used to define a class of objects by listing the prohibited subobjects, i.e., subobjects that are not allowed to be contained in the objects of the class. The notion of a subobject is defined in different ways depending on the objects under consideration: a subword (a block or segment) for fragmentarily restricted languages, a subgraph for families of graphs, a subshape for two-dimensional shapes (e.g., a submatrix for matrices), and so on.

We collect all prohibited objects into a set that we call a set of prohibited objects, or simply a *set of prohibitions*. The idea of *unavoidable* (or *complete*[1]) set is as follows: if large enough objects must contain prohibited subobjects, then the set of prohibitions is unavoidable.

In this paper, we are interested in unavoidable sets of *word patterns*, or just *patterns* (see section 3 for definitions). These patterns are an extension of the *permutation patterns* studied extensively for the last twenty years (see [17] for a survey on the corresponding problems). Our unavoidable sets of patterns are refinements for those of words. Questions on unavoidability of sets of words appear, for instance, in algebra (sequences without repetitions), coding theory (chain codes), number theory (arithmetic progressions in partitions of the set of natural numbers and, e.g., van der Waerden's theorem), and dynamical systems (motions of an object in a space with certain restrictions).

There are a number of numerical characteristics that are valuable for unavoidability criteria and the recognition algorithms based on them. Three such characteristics,

*Received by the editors August 18, 2004; accepted for publication (in revised form) February 4, 2005; published electronically October 31, 2005.
http://www.siam.org/journals/sidma/19-2/44567.html

[†] Department of Mathematics, Iowa State University, Ames, IA 50011-2064 (burstein@orion.math.iastate.edu).

[‡] Department of Mathematics, University of Kentucky, Lexington, KY 40506-0027 (kitaev@ms.uky.edu).

[1] The word *complete* appears, for example, in [6, 7, 8, 9, 10], but the word *unavoidable* is of common use in contemporary literature (e.g., see [1], [19, Chapter 3], [21]), so we decided to use the latest terminology in this paper. Another expression in the literature for unavoidable sets is *blocking sets* [22].

namely, $M_w(n)$, $L_w(n)$, and $C_w(n)$ (for definitions see section 2), are considered in [7]. We consider the similar characteristics $M_p(n,m)$, $L_p(n,m)$, and $C_p(n,m)$ for the case of prohibited patterns (for definitions see section 3), where m is the number of letters in the corresponding alphabet (we do not use this parameter for the functions $M_w(n)$, $L_w(n)$, and $C_w(n)$ to be consistent with [7]). Moreover, in section 3.2.2 we discuss how finding a lower bound for $C_p(n,m)$ is related to the so-called *universal cycles for combinatorial structures* that have been studied extensively (e.g., see [5, 16] and the references therein). To get the lower bound involving the Stirling numbers of the second kind, we prove that the *graph of pattern overlaps* (see section 3) is Hamiltonian and derive as a corollary that there exists a universal cycle for word patterns of any length over any alphabet (see Corollary 3.9).

We remark that when considering patterns, the underlying alphabet must be ordered, as opposed to the objects considered in [7].

The paper is organized as follows. In section 2 we review the main results on unavoidable sets of words in [7, 8]. The motivation for a relatively detailed review of these papers is the fact that they are available only in Russian (as far as we know), which caused, in particular, the rediscovery of some of those results in [21]. In addition, the results obtained in [7, 8] are of great interest in general and very useful in this paper in particular. In section 3, we define the notion of a *pattern*, an *n-pattern word*, and study unavoidable sets of patterns.

2. Unavoidable sets of words. Let $\mathcal{A} = \{a_1, \dots, a_n\}$ be an alphabet of n letters. A *word* over the alphabet $\mathcal{A}$ is a finite sequence of letters of the alphabet. Any i consecutive letters of a word X generate a *subword* of length i. The set $\mathcal{A}^*$ is the set of all words over the alphabet $\mathcal{A}$, and $\mathcal{A}^n$ is the set of all words over $\mathcal{A}$ of length n. Let $S \subseteq \mathcal{A}^*$ be a set of prohibited words or a set of prohibitions. A word that does not contain any words from S as its subwords is said to be *free* from S, or S-*free*. The set of all S-free words is denoted by $\widehat{S}$.

If there exists a natural number k such that the length of any word in $\widehat{S}$ is less than k, then S is called an unavoidable set. It is straightforward to see that S is unavoidable if and only if $\widehat{S}$ has finitely many of elements. Thus, for any unavoidable set S we can define the function

$$L_w(\widehat{S}) = \max_{X \in \widehat{S}} \ell(X),$$

where $\ell(X)$ is the length of a word X.

The basic problem in considering sets of prohibitions is whether or not a given set S of prohibitions is unavoidable. Other possible problems include the following: given an unavoidable S, find or estimate $L_w(\widehat{S})$; construct an S-free word of length $L_w(\widehat{S})$; find the number of elements in $\widehat{S}$. If S is avoidable, then some possible problems could include the following: find an infinite S-free sequence; describe all such sequences; find the cardinality of the set of these sequences; find the cardinality of the set of finite S-avoiding sequences of a given length.

Let S be a finite set of words over an alphabet $\mathcal{A}$, and let n be the maximal length of a word in S. If X is a subword of Y, then we say that Y is a *superword* for X. Suppose now that $X \in S$ and $\ell(X) < n$. Remove X from S and adjoin to S all superwords for X of length n. If this procedure is performed for any such X, and all resulting repetitions are removed, we will get a set $S^{'}$ of distinct words of length n.

PROPOSITION 2.1 (see [7, Proposition 1]). *A set S is unavoidable if and only if $S^{'}$ is unavoidable.*

Thus, sets of prohibitions $S \subseteq \mathcal{A}^n$ are of special interest, and for the most part, our considerations in this paper are related to these sets. More precisely, we will consider the functions

$$M_w(n) = \min |S| \quad \text{and} \quad L_w(n) = \max L_w(\widehat{S}),$$

where the extremum is taken with respect to all unavoidable $S \subseteq \mathcal{A}^n$. These functions are examples of numerical characteristics that describe the bound between avoidable and unavoidable sets of prohibitions. To give an instance of such a bound, we consider the following example.

Example 2.2 (see [7, Examples 1, 2]). *Consider $\mathcal{A} = \{0,1\}$ and the sets of prohibitions*

$$\begin{aligned} S_1 &= \{000, 001, 101\underline{1}, 0101, 1111\}, \\ S_2 &= \{000, 001, 101\underline{0}, 0101, 1111\}. \end{aligned}$$

Thus S_1 and S_2 differ only in one letter (underlined). One can see that S_1 is unavoidable, and $L_w(\widehat{S_1}) = 8$. On the other hand, S_2 is avoidable. Indeed,

$$\underbrace{011}\,\underbrace{011}\ldots \quad \textit{and} \quad \underbrace{0111}\,\underbrace{0111}\ldots$$

are S_2-free, and

$$\underbrace{011}\,\underbrace{0111} \quad \textit{and} \quad \underbrace{0111}\,\underbrace{011}$$

are S_2-free. Hence, substituting $0 \mapsto 011$ and $1 \mapsto 0111$ in any sequence over $\mathcal{A}$, we get an S_2-free sequence. Hence, the cardinality of $\widehat{S_2}$ is the continuum.

In what follows, we will need the following graph. A *de Bruijn graph* is a directed graph $\vec{G}_n = \vec{G}_n(V, E)$, where the set of vertices V is the set of all words in $\mathcal{A}^n$, and there is an arc from $u \in \mathcal{A}^n$ to $v \in \mathcal{A}^n$ if and only if

$$u = aw \text{ and } v = wb \quad \text{for some } w \in \mathcal{A}^{n-1} \text{ and } a, b \in \mathcal{A}.$$

Figure 1 shows the de Bruijn graphs for a 2-letter alphabet and $n = 2, 3$.

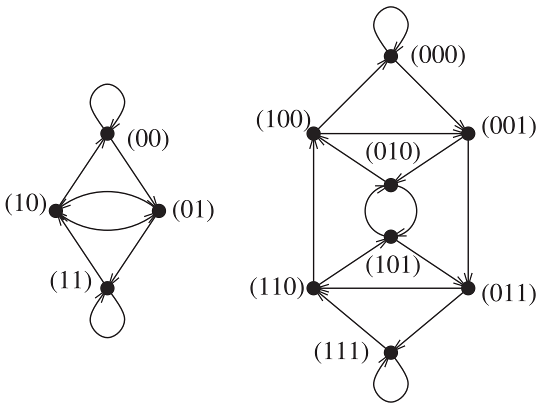


FIG. 1. *The de Bruijn graphs for the alphabet $\mathcal{A} = \{0,1\}$ and $n = 2, 3$.*

The de Bruijn graphs were first introduced (for the alphabet $\mathcal{A} = \{0,1\}$) by de Bruijn in 1944 for finding the number of code cycles. However, these graphs proved

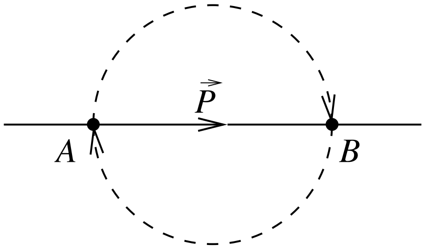


FIG. 2. *The arc $\vec{BA}$ is a chord for the path $\vec{P}$, but $\vec{AB}$ is not.*

to be a useful tool for various problems related to combinatorics on words (e.g., see [7, 8, 11, 12, 14, 15]). It is known that the graph $\vec{G}_n$ can be defined recursively as $\vec{G}_n = L(\vec{G}_{n-1})$, where L indicates the operation of taking the line graph.

A *chord* of a directed simple path $\vec{P}$ in $\vec{G}_n$ is an arc that does not belong to $\vec{P}$ but connects two of its vertices in a such way that there is a circuit generated by this arc and the part of the path between the ends of the arc. For instance, on Figure 2 the arc $\vec{BA}$ is a chord for the path $\vec{P}$, whereas $\vec{AB}$ is not.

Let $C_w(n)$ denote the greatest length (the number of vertices) of a simple path in $\vec{G}_n$ that does not have chords and does not go through any vertex that has a loop. The following theorem was proved by considering the de Bruijn graph.

THEOREM 2.3 (see [7, Theorem 1]). *We have $L_w(n) = C_w(n) + n - 1 = |\mathcal{A}|^{n-1} + n - 2$.*

The following theorem was proved using the cyclic structure of the de Bruijn graph (the main result of [20]) as well as the number of conjugacy classes of words with respect to a cyclic shift.

THEOREM 2.4 (see [7, Theorem 2]). *We have*

$$M_w(n) = \frac{1}{n}\sum_{d|n} \varphi(n/d)|\mathcal{A}|^d,$$

where $\varphi(n)$ is the number of integers in $\{1, 2, \dots, n-1\}$ relatively prime to n (Euler's φ-function).

Since any set of prohibitions S with $|S| < M_w(n)$ is avoidable, it is helpful to have a table for $M_w(n)$. For $|\mathcal{A}| = 2$ and $2 \le n \le 10$, see Table 1.

TABLE 1
The function $M_w(n)$ for $2 \le n \le 10$ and a 2-letter alphabet.

n	2	3	4	5	6	7	8	9	10
$M_w(n)$	3	4	6	8	14	20	36	60	108

In particular, any set of binary words of length 9 that has fewer than 60 words is avoidable. Also, it is obvious that $M_w(n) \sim |\mathcal{A}|^n/n$, when $n \to \infty$. The last observation allows us to prove the following statement.

PROPOSITION 2.5 (see [8, Proposition 1]). *There exist at least $2^{|\mathcal{A}|^n(1-\varepsilon_n)}$ unavoidable sets $S \subseteq \mathcal{A}^n$. Here $\varepsilon_n \to 0$ when $n \to \infty$.*

3. Unavoidable sets of patterns. The alphabets considered in this section must be totally ordered, and without loss of generality they coincide with $[m] = \{1, 2, \dots, m\}$ for an appropriate m.

We refer to [2, 17] for a general introduction and survey of various pattern problems. However, in this paper we are concerned only with *word patterns* studied for the first time in [3]. More precisely, we consider the *segment* word patterns (see [17]) (also referred to as generalized patterns *without internal dashes*), i.e., those whose occurrence must be a string of consecutive letters. For this paper, we can define a pattern to be a subword (of a word) that contains each of the letters $1, 2, \ldots, k$ at least once for some k, and no other letters. For instance, the word 2613235 contains an occurrence of the pattern 1323, but its subword 2613 is not a pattern. By analogy with section 2, if a word does not contain a pattern p, it is *free* from p, or *p-free.* However, the crucial difference between this section and section 2 is that instead of considering *words* free from a pattern p, we consider the objects that we call the *n-pattern words.* An n-pattern word is a word in which each subword of length n is a pattern. Thus, in the construction of n-pattern words, we can restrict ourselves to alphabets having at most n letters. Indeed, an occurrence of a letter $m > n$ in a subword A of length n of an n-pattern word W contradicts the fact that A must be a pattern (A must contain each of the letters $1, 2, \ldots, m$).

By analogy with section 2, when dealing with sets of prohibited words, we can consider sets of prohibited patterns, or simply sets of prohibitions, when it is clear which prohibitions we mean. We can also define the notion of an unavoidable set here in the same way. However, in considering prohibited patterns and n-pattern words, we assume that all prohibitions are of length n. Hence, for patterns, we can define the functions $L_p(n, m)$ and $M_p(n, m)$ similarly to $L_w(n)$ and $M_w(n)$ (recall that m is the number of letters in the alphabet). As in section 2, the basic problem is whether or not a given set S_p of prohibitions is unavoidable, and $L_p(n, m)$ and $M_p(n, m)$ are important numerical characteristics to study.

3.1. The function $M_p(n, m)$. Recall that the Möbius function is defined by

$$\mu(n) = \begin{cases} 0 & \text{if } n \text{ has one or more repeated prime factors,} \\ 1 & \text{if } n = 1, \\ (-1)^k & \text{if } n \text{ is a product of } k \text{ distinct primes,} \end{cases}$$

so $\mu(n) \neq 0$ indicates that n is square-free.

The purpose of this subsection is to prove the following theorem.

THEOREM 3.1. *For n-pattern words over $[m]$, we have*

$$M_p(n, m) = \sum_{i|n} \sum_{j=0}^{\min(i,m)-1} (-1)^j \binom{\min(i, m) - 1}{j} \frac{1}{i} \sum_{d|i} \mu(d)(\min(i, m) - j)^{\frac{i}{d}},$$

where $M_p(n, m) = \min |S_p|$, and the minimum is taken over all unavoidable sets S_p of patterns of length n over the alphabet $[m]$.

One can compare this result with that of Theorem 2.4.

Remark 3.2. In Theorem 3.1, we can assume that $n \geq m$, since if $n < m$, we can use only the first n letters in $[m]$ to construct n-pattern words, which reduces to the case $n = m$.

Remark 3.3. For $n = m$, we have $\min(i, m) = i$ in the formula of Theorem 3.1.

To prove Theorem 3.1, we introduce the *graph of pattern overlaps* $\vec{P}_n = \vec{P}_n(V, E)$, which is a subgraph of the de Bruijn graph $\vec{G}_n$, where the set of vertices V contains all n-letter patterns over the underlying alphabet $\mathcal{A}$, and the set of arcs E consists

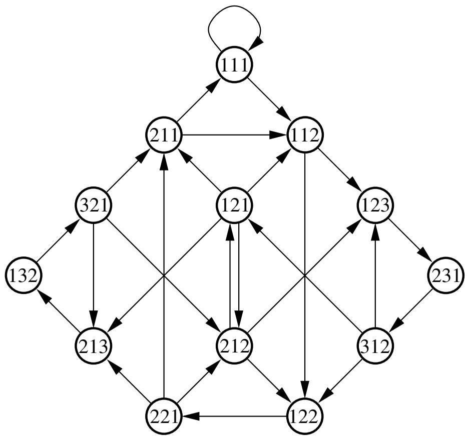


FIG. 3. *The graph of pattern overlaps for $\mathcal{A} = \{1, 2, 3\}$ and $n = 3$.*

of all the arcs of $\vec{G}_n$ between vertices corresponding to the patterns. In Figure 3, we can see the graph of pattern overlaps in the case of a 3-letter alphabet and $n = 3$ (we omit parentheses around the triples on the graph to indicate that we are dealing with $\vec{P}_3$, not $\vec{G}_3$).

Let $T_p(n, m)$ denote the number of conjugacy classes of patterns of length n over the alphabet $[m]$ with respect to a cyclic shift. For instance, there are 5 conjugacy classes on Figure 3. They are $\{111\}$, $\{112, 121, 211\}$, $\{221, 212, 122\}$, $\{321, 213, 132\}$, and $\{312, 123, 231\}$. Thus, $T_p(3, 3) = 5$.

LEMMA 3.4. *We have $M_p(n, m) = T_p(n, m)$.*

Proof. To prove the lemma, we follow the proof of Theorem 2.4 in [7].

Suppose S_p is an unavoidable set of patterns of length n and X is an arbitrary n-pattern word of length n (X is a pattern) over $[m]$. We form the sequence

$$X^\infty = XXX\dots$$

by repeating the word X periodically. Since S_p is unavoidable, X^∞ contains a prohibited pattern $p \in S_p$. From the construction of the sequence, p is either X or a cyclic shift of X. Thus S_p contains a pattern from each conjugacy class of patterns of length n over $[m]$ with respect to a cyclic shift. Thus, $|S_p| \geq T_p(n, m)$, and since S_p is an arbitrary set, we have

$$M_p(n, m) \geq T_p(n, m).$$

To prove that $T_p(n, m)$ is an upper bound, we need to find an unavoidable set of cardinality $T_p(n, m)$. We consider the graph $\vec{P}_n$ whose vertices correspond to the words over $[m]$. If $V' \subset V(\vec{P}_n)$ and each circuit of $\vec{P}_n$ contains a vertex in V', then we say that V' *cuts* all circuits of $\vec{P}_n$. By deleting all such V' with all incident arcs from $\vec{P}_n$, we get an acyclic graph on the vertex set $V \backslash V'$. The set of the patterns in $[m]^n$ corresponding to the vertices in V' is unavoidable. Indeed, if not, a sequence free from V' determines a self-intersecting walk in $\vec{P}_n$ and thus generates a circuit on the vertex set $V \backslash V'$, which is impossible.

Mykkeltveit [20] found a set of vertices V_c that cuts all circuits of the de Bruijn graph $\vec{G}_n$ with $|V_c|$ equal to the number of conjugacy classes of the words. Thus

V_c cuts all circuits in $\vec{G}_n$ and has one vertex in each conjugacy class. Since $\vec{P}_n$ is a subgraph of $\vec{G}_n$, $\vec{P}_n$ will have no circuit after removing the vertices in V_c. The set of vertices in V_c that belong to $\vec{P}_n$ corresponds to an unavoidable set, and thus

$$M_p(n,m) \le T_p(n,m).$$

This proves the lemma. □

Lemma 3.5.

$$T_p(n,m) = \sum_{i|n} \sum_{j=0}^{\min(i,m)-1} (-1)^j \binom{\min(i,m)-1}{j} \frac{1}{i} \sum_{d|i} \mu(d)(\min(i,m)-j)^{\frac{i}{d}}.$$

Proof. Recall that a word $x \in \mathcal{A}^*$, where $\mathcal{A}$ is any (ordered or unordered) alphabet, is called *primitive* if it is not a power of another word. Thus $x \neq \emptyset$ is primitive if $x = y^e$ only for $e = 1$. For instance, the words 121, 1221, 12121 are primitive, whereas the word 121212 is not. It is easy to show that each nonempty word is a power of a unique primitive word. Thus, $x = r^e$ for a unique primitive word r. The number e is called the *exponent* of x. It is also easy to see that all words, and hence all patterns, in the same conjugacy class have the same exponent. Moreover, if $x_1 = r_1^e$ and $x_2 = r_2^e$ and $|x_1| = |x_2|$, then x_1 is conjugate to x_2 if and only if r_1 is conjugate to r_2. We define the notion of a *primitive pattern* in the same way as for words. Clearly, all properties of primitive words hold for primitive patterns as well.

So, in order to find $T_p(n,m)$, we need to find the number of conjugacy classes of primitive patterns of length i over the alphabet $[m]$, where $i|n$, and then take a sum of these numbers. However, for a given i, we cannot use directly the well-known formula for the number of conjugacy classes of primitive words over $\min(i,m)$-letter alphabet (a primitive word of length i can have at most i distinct letters, since we are dealing with patterns), given by

$$\frac{1}{i} \sum_{d|i} \mu(d)(\min(i,m))^{\frac{i}{d}}.$$

Indeed, this formula counts, among others, primitive words which are not primitive patterns (when some letter j, $2 \le j \le \min(i,m)-1$, occurs in a primitive pattern, whereas $j-1$ does not). So, we need to use the standard inclusion-exclusion method (the sieve formula) to handle this situation. We define the property A_j to be "the letter j does not occur in a primitive word." Clearly we may restrict ourselves to the case $j \le \min(i,m)-1$, since the absence of the largest letter, namely, $\min(i,m)$, is not a bad property when considering patterns. Now we easily get the number of primitive patterns of length i, which is given by

$$\sum_{j=0}^{\min(i,m)-1} (-1)^j \binom{\min(i,m)-1}{j} \frac{1}{i} \sum_{d|i} \mu(d)(\min(i,m)-j)^{\frac{i}{d}}.$$

This proves the lemma. □

Now the truth of Theorem 3.1 follows from Lemmas 3.4 and 3.5.

3.2. The function $L_p(n,m)$. Let $C_p(n,m)$ denote the greatest length (the number of vertices) of a simple path in $\vec{P}_n$ that does not have chords (see the definition in section 2) and does not pass through any vertex incident with a loop. Using exactly the same considerations as in the proof of Theorem 2.3 (see [7]), one can prove the following theorem.

THEOREM 3.6. *We have* $L_p(n, m) = C_p(n, m) + n - 1$.

Moreover, in the case $m = 2$, the de Bruijn graph $\vec{G}_n$ almost coincides with the graph of pattern overlaps $\vec{P}_n$. Indeed, the only difference between these graphs is the vertex $(22 \dots 2)$ and all edges adjacent to that vertex ($22 \dots 2$ is the only binary nonpattern). However, the lemma to Theorem 2.3 (see [7]) provides that in the binary case $C_w(n) = 2^{n-1} - 1$, and since $C_w(n)$ is the maximal length of a path that, in particular, does not pass through the loop $(22 \dots 2)$, we have that in this case $C_w(n) = C_p(n, 2)$. Thus the following theorem is true.

THEOREM 3.7. *We have* $L_p(n, 2) = 2^{n-1} + n - 2$.

However, in the case $m \geq 3$, the only useful information we can extract from Theorem 2.3 is the following rough bound:

$$L_p(n, m) < m^{n-1} + n - 2.$$

So, according to Theorem 3.6 we need to find $C_p(n, m)$ in order to get $L_p(n, m)$. The purpose of the rest of the subsection is to find an upper and a lower bound for $C_p(n, m)$ for $m \geq 3$.

3.2.1. An upper bound for $C_p(n, m)$. We give only a trivial upper bound. Clearly, in order to avoid chords, each conjugacy class (with respect to shift) which has i words can have no more than $i - 1$ words in the path. Thus, we use the formula for $T_p(m, n)$ with a correction, namely, the factor of $i - 1$, which indicates that each primitive word of length i is responsible for a conjugacy class of i elements, and we take $i - 1$ elements out of these i:

$$C_p(n, m) \leq \sum_{i|n} (i - 1) \sum_{j=0}^{\min(i,m)-1} (-1)^j \binom{\min(i, m) - 1}{j} \frac{1}{i} \sum_{d|i} \mu(d)(\min(i, m) - j)^{\frac{i}{d}}.$$

3.2.2. A lower bound for $C_p(n, m)$. We observe that the line graph $L(\vec{P}_{n-1})$ for the graph $\vec{P}_{n-1}$ determines a subgraph of the graph $\vec{P}_n$. We get that by using the general properties of the de Bruijn graph (since $\vec{P}_n$ is its subgraph), as well as the fact that if $x_1 x_2 \dots x_{n-1}$ and $x_2 x_3 \dots x_n$ are vertices in $\vec{P}_{n-1}$, then the arc between them generates the vertex $x_1 x_2 \dots x_n$ in the line graph, and $x_1 x_2 \dots x_n$ is a pattern and thus belongs to $\vec{P}_n$. Moreover, from the considerations in the proof of Theorem 2.3 (see [7]), it follows that a simple path in $\vec{P}_{n-1}$ determines a simple path without chords in $\vec{P}_n$ after removing the loop $11 \dots 1$.

So, in order to get a lower bound for $C_p(n, m)$, we need to construct a simple path in $\vec{P}_{n-1}$ of as great a length as possible (ideally a Hamiltonian path). In order to get a Hamiltonian path or a path that is "close" to a Hamiltonian one, we can try to use the methods and techniques similar to those used in constructions of universal cycles for various combinatorial structures such as words, permutations, partitions, and others (e.g., see [5, 16]).

We briefly discuss the general notion of a universal cycle (see [5]).

Suppose we are given a family $\mathcal{F}_n$ of combinatorial objects of "rank n" and let $m := |\mathcal{F}_n|$ denote their number. We assume that each $F \in \mathcal{F}_n$ is "generated" or specified by some sequence $x_1 x_2 \dots x_n$, where $x_i \in \mathcal{A}$ for some fixed alphabet $\mathcal{A}$. We say that $U = a_0 a_1 \dots a_{m-1}$ is a universal cycle (or a *U-cycle*) for $\mathcal{F}_n$ if $a_{i+1} a_{i+2} \dots a_{i+n}$,

$0 \le i < m$, runs through each element of $\mathcal{F}_n$ exactly once, where index addition is performed modulo n.

In our case the combinatorial objects are patterns of length n, and as in many other cases (e.g., *de Bruijn cycles*, permutations, partitions), but not in all cases (e.g., k-subsets of an n-set), it is possible to define a directed *transition graph*, namely, the graph of pattern overlaps $\vec{P}_n$, and reduce the problem of constructing a U-cycle to constructing a Hamiltonian circuit for $\vec{P}_n$. Even though we do not need a Hamiltonian circuit (since we are concerned with paths of maximal length), we can still try to use the same techniques as in [5, 16] and in the references therein.

However, it turns out that the above-mentioned techniques work only for $m = 2$, which we are not interested in since we have an explicit result in this case (see Theorem 3.7). The main problem is that the graph of pattern overlaps is not balanced; i.e., we have vertices where the indegree is not equal to the outdegree. Also, $\vec{P}_n$ is not the line graph of $\vec{P}_{n-1}$. However, it is possible to prove the following statement.

THEOREM 3.8. *The graph of pattern overlaps $\vec{P}_n$ contains a Hamiltonian circuit.*

Proof. We first observe that $\vec{P}_n$ is connected. Indeed, suppose we are given two vertices of $\vec{P}_n$, namely, $X = x_1x_2\dots x_n$ and $Y = y_1y_2\dots y_n$. If I denotes the vertex $11\dots1$, then we can find a path $\vec{P}_X$ from X to I. Indeed, if x_i is the largest letter in X, then we consider the following path in $\vec{P}_n$:

$$X = x_1x_2\dots x_n \to x_2x_3\dots x_nx_1 \to \cdots \to x_ix_{i+1}\dots x_{i-1} \to x_{i+1}\dots x_{i-1}1 = X'.$$

Thus, in X' we get 1 in place of the largest letter of X. We observe that X' is obviously a pattern. Clearly, we can continue this path by replacing the largest letters, one by one, with 1's until we arrive at I. On the other hand, it is easy to see that the operation of changing a largest letter to 1 is invertible. For instance, in order to find a path from X' to X, we may perform the following sequence of steps:

$$\begin{aligned}X' = x_{i+1}\dots x_{i-1}1 \to x_{i+2}\dots 1x_{i+1} \to \cdots \to 1x_{i+1}\dots x_nx_1\dots x_{i-1} \to\\ x_{i+1}\dots x_nx_1\dots x_{i-1}x_i \to x_{i+2}\dots x_ix_{i+1} \to \cdots \to x_1x_2\dots x_n = X.\end{aligned}$$

Thus, we can find a path from I to Y, which together with the path $\vec{P}_X$ gives a path from X to Y. Similarly, one can get a path from Y to X, which proves that $\vec{P}_n$ is connected.

We now use standard line digraph methods to finish our proof.

Let $\vec{D}_{n-1}$ be the subgraph of the de Bruijn graph $\vec{G}_{n-1}$ with the same vertex set as $\vec{G}_{n-1}$ and arcs corresponding to vertices of $\vec{P}_n$. So the line digraph of $\vec{D}_{n-1}$ is $\vec{P}_n$. We need to show that $\vec{D}_{n-1}$ is Eulerian up to isolated vertices, which would imply that $\vec{P}_n$ is Hamiltonian.

The degree condition for $\vec{D}_{n-1}$ is satisfied since vertices corresponding to patterns with k letters will have in- and outdegrees equal to the minimum of $k+1$ and the alphabet size; vertices that are almost patterns (consisting of the letters $\{1, 2, \dots, i-1, i+1, \dots, j\}$ for some i and j) have in- and outdegree 1; other vertices are isolated.

Connectivity for $\vec{D}_{n-1}$ (except for the isolated vertices) can be obtained from connectivity for $\vec{P}_n$ proved above. Indeed, for any two nonisolated vertices x and y in $\vec{D}_{n-1}$ there are arcs X and Y such that X comes out from x and Y comes in to y. There is a path from X to Y in $\vec{P}_n$ which gives a path from x to y in $\vec{D}_{n-1}$. Similarly, there is a path from y to x. We are done. ☐

As an immediate corollary to Theorem 3.8 we have the following.

Corollary 3.9. *For any m and n, there exists a U-cycle for word patterns of length n over an m-letter alphabet.*

The following proposition is easy to prove using elementary combinatorics.

Proposition 3.10. *The number of different word patterns of length n on m letters is*

$$\sum_{i=1}^{m} \sum_{\substack{a_1+\cdots+a_i=n\\ a_1\geq 1,\ldots,a_i\geq 1}} \binom{n}{a_1,\ldots,a_i} = \sum_{i=1}^{m} i!S(n,i),$$

where $S(n,i)$ is a Stirling number of the second kind.

Now, using the discussion in the beginning of section 3.2.2, Theorem 3.8, and Proposition 3.10, we obtain the following proposition, where again $S(n,i)$ is a Stirling number of the second kind.

Proposition 3.11.

$$C_p(n,m) \geq \sum_{i=1}^{m} \sum_{\substack{a_1+\cdots+a_i=n-1\\ a_1\geq 1,\ldots,a_i\geq 1}} \binom{n-1}{a_1,\ldots,a_i} = \sum_{i=1}^{m} i!S(n-1,i).$$

As a final remark, we observe that another way to get the number of different word patterns of length n on m letters is using a correction in the formula for $T_p(m,n)$ just as we did when we obtained the upper bound for $C_p(n,m)$ in section 3.2.1. However, in this case the correction factor is i rather than $i-1$, which says that we consider each conjugacy class with respect to shift and find the number of elements in it. Thus, i and $1/i$ cancel each other, and we get a combinatorial proof of the following identity:

$$\sum_{i=1}^{m} i!S(n,i) = \sum_{i|n} \sum_{j=0}^{\min(i,m)-1} (-1)^j \binom{\min(i,m)-1}{j} \sum_{d|i} \mu(d)(\min(i,m)-j)^{\frac{i}{d}}.$$

Acknowledgment. The authors would like to thank the anonymous referee for bringing references [11, 15] to their attention, as well as for improved drawings of Figures 2 and 3.

REFERENCES

[1] D. Bean, A. Ehrenfeucht, and G. McNulty, *Avoidable patterns in strings of symbols*, Pacific J. Math., 85 (1979), pp. 261–294.

[2] M. Bóna, *Combinatorics of Permutations*, Chapman & Hall/CRC, Boca Raton, FL, 2004.

[3] A. Burstein and T. Mansour, *Words restricted by patterns with at most 2 distinct letters*, Electron. J. Combin., 9 (2002/2003), article R3.

[4] C. Choffrut and J. Karhumäki, *Combinatorics of words*, in Handbook of Formal Languages, Vol. 1: Word, Language, Grammar, Springer, Berlin, 1997, pp. 329–438.

[5] F. Chung, P. Diaconis, and R. Graham, *Universal cycles for combinatorial structures*, Discrete Math., 110 (1992), pp. 43–60.

[6] A. Evdokimov, *Completeness of a Word Set*, Talk at the international conference FCT-79, Wendisch Rietz, Germany, 1979.

[7] A. Evdokimov, *Complete sets of words and their numerical characteristics*, Metody Diskret. Analiz., 39 (1983), pp. 7–32 (in Russian).

[8] A. Evdokimov, *The completeness of sets of words*, in Proceedings of the All-Union Seminar on Discrete Mathematics and Its Applications (Moscow, 1984), Moskov. Gos. Univ., Mekh.-Mat. Fak., Moscow, 1986, pp. 112–116 (in Russian).

[9] A. Evdokimov and S. Kitaev, *Crucial words and the complexity of some extremal problems for sets of prohibited words*, J. Combin. Theory Ser. A, 105 (2004), pp. 273–289.

[10] A. Evdokimov and V. Krainev, *Problems on completeness of sets of words*, in Proceedings of the 22nd Regional Scientific Conference by the Popov Association, Novosibirsk, 1979, pp. 105–107 (in Russian).

[11] C. Flye-Sainte Marie, *Solution to problem number* 58, l'Intermediaire de Mathematiciens, 1 (1894), pp. 107–110.

[12] H. Fredricksen, *A survey of full length nonlinear shift register cycle algorithms*, SIAM Rev., 24 (1982), pp. 195–221.

[13] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman, San Francisco, CA, 1979.

[14] S. W. Golomb, *Shift Register Sequences*, Holden-Day, San Francisco, CA, 1967.

[15] I. J. Good, *Normally recurring decimals*, J. London Math. Soc., 21 (1946), pp. 167–169.

[16] G. Hurlbert, *Universal Cycles: On Beyond de Bruijn*, Ph.D. thesis, Department of Mathematics, Rutgers University, Piscataway, NJ, 1990.

[17] S. Kitaev and T. Mansour, *A Survey of Certain Pattern Problems*, preprint.

[18] M. Lothaire, *Combinatorics on Words*, Encyclopedia Math. Appl. 17, Addison-Wesley, Reading, MA, 1983.

[19] M. Lothaire, *Algebraic Combinatorics on Words*, Encyclopedia Math. Appl. 90, Cambridge University Press, Cambridge, UK, 2002.

[20] J. Mykkeltveit, *Proof of Golomb's conjecture for the de Bruijn graph*, J. Combin. Theory Ser. B, 13 (1972), pp. 40–45.

[21] C. Saker and P. Higgins, *Unavoidable sets of words of uniform length*, Inform. and Comput., 173 (2002), pp. 222–226.

[22] A. Zimin, *Blocking sets of terms*, Mat. Sb. (N.S.), 119 (1982), pp. 363–375, 447 (in Russian); Math. USSR-Sb., 47 (1984), pp. 353–364 (in English).

SIAM J. DISCRETE MATH.
Vol. 19, No. 2, pp. 382–387

A CONVEX QUADRATIC CHARACTERIZATION OF THE LOVÁSZ THETA NUMBER*

CARLOS J. LUZ† AND ALEXANDER SCHRIJVER‡

Abstract. In previous works an upper bound on the stability number $\alpha(G)$ of a graph G based on convex quadratic programming was introduced and several of its properties were established. The aim for this investigation is to relate theoretically this bound (usually represented by $\upsilon(G)$) with the well-known Lovász $\vartheta(G)$ number. First, a new set of convex quadratic bounds on $\alpha(G)$ that generalize and improve the bound $\upsilon(G)$ is proposed. Then it is proved that $\vartheta(G)$ is never worse than any bound belonging to this set of new bounds. The main result of this note states that one of these new bounds equals $\vartheta(G)$, a fact that leads to a new characterization of the Lovász theta number.

Key words. combinatorial optimization, graph theory, maximum stable set, quadratic programming

AMS subject classifications. 05C50, 68R10, 90C20, 90C27

DOI. 10.1137/S0895480104429181

1. Introduction.

Let $G = (V, E)$ be a simple undirected graph where $V = \{1, \dots, n\}$ denotes the vertex set and E is the edge set. It will be supposed that G has at least one edge, i.e., E is not empty. We will write $ij \in E$ to denote the edge linking nodes i and j of V. The adjacency matrix $A_G = [a_{ij}]$ of G is defined by

$$a_{ij} = \begin{cases} 1 & \text{if } ij \in E, \\ 0 & \text{if } ij \notin E. \end{cases}$$

A stable set (independent set) of G is a subset of nodes of V whose elements are pairwise nonadjacent. The stability number (or independence number) of G is defined as the cardinality of a largest stable set and is usually denoted by $\alpha(G)$. A maximum stable set of G is a stable set with $\alpha(G)$ nodes. The problem of finding $\alpha(G)$ is NP-hard, and thus it is suspected that it cannot be solved in polynomial time. In addition, there exists $\epsilon > 0$ such that to approximate $\alpha(G)$ within a ratio of $n^{-\epsilon}$ is NP-hard (see [1]). However, several ways of approaching $\alpha(G)$ have been proposed in the literature (see, for example, [2, 6, 9, 16] and [3] for a survey).

For any graph G with at least one edge, it can easily be proved (see Proposition 2.1) that $\alpha(G) \le \upsilon(G)$, where $\upsilon(G)$ is the optimal value of the following convex quadratic programming problem:

$$(P_G) \qquad \upsilon(G) = \max\{2e^T x - x^T (H + I) x : x \ge 0\}.$$

Here and hereinafter e is the $n \times 1$ all ones vector, T stands for the transposition operation, I is the identity matrix of order n, and

$$H = \frac{1}{-\lambda_{\min}(A_G)} A_G,$$

*Received by the editors June 11, 2004; accepted for publication (in revised form) December 27, 2004; published electronically October 31, 2005.
http://www.siam.org/journals/sidma/19-2/42918.html

†Escola Superior de Tecnologia de Setúbal/Instituto Politécnico de Setúbal, Campus do IPS, Estefanilha, 2914-508 Setúbal, Portugal (cluz@est.ips.pt).

‡CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands (lex@cwi.nl).

where A_G is the adjacency matrix of G and $\lambda_{\min}(A_G)$ is its smallest eigenvalue. As the trace of A_G is zero and G has at least one edge, A_G is indefinite. (See [5] for details.) Thus $\lambda_{\min}(H) = -1$ and this guarantees the convexity of P_G because $H + I$ is positive semidefinite.

Having in mind the nice properties of $\upsilon(G)$ (see [14, 15]), the initial aim of this investigation was to theoretically relate $\upsilon(G)$ with the well-known Lovász $\vartheta(G)$ number introduced in [12] and discussed in many publications [8, 9, 10, 11, 13]. As a consequence of this effort, a new set of convex quadratic bounds on $\alpha(G)$ that generalize and improve the $\upsilon(G)$ bound is introduced. Also, it is shown that $\vartheta(G)$ is never worse than any bound belonging to this set of new bounds. The main result herein proved states that $\vartheta(G)$ is equal to the best bound in this set. In consequence, it leads to a characterization of $\vartheta(G)$ by convex quadratic programming.

This note is organized as follows. In section 2 the new family of upper bounds on $\alpha(G)$ is introduced and some of its properties are presented. In section 3, some different $\vartheta(G)$ formulations are recalled, and the results relating the new introduced bounds with $\vartheta(G)$ are established in section 4.

2. Generalizing the $\upsilon(G)$ bound.

To improve the upper bound $\upsilon(G)$, we define the following family of quadratic problems which are based on a perturbation in the Hessian of the convex quadratic programming problem P_G:

$$(P_G(C)) \qquad \upsilon(G,C) = \max\{2e^T x - x^T (H_C + I)\, x : x \geq 0\},$$

where $C = [c_{ij}]$ is a nonnull real symmetric matrix such that $c_{ij} = 0$ if $i = j$ or $ij \notin E$ and

$$H_C = \frac{C}{-\lambda_{\min}(C)},$$

denoting $\lambda_{\min}(C)$ the smallest eigenvalue of C. Any matrix satisfying the conditions imposed to matrix C will be called a *weighted adjacency matrix* of G. Note that as well as the adjacency matrix A_G, the matrix C is indefinite, taking into account that its trace is null and not all entries c_{ij} are null. Consequently, since $\lambda_{\min}(H_C) = -1$, all problems $P_G(C)$ are convex. Note also that $\upsilon(G, A_G) = \upsilon(G)$ and thus P_G is included in the introduced family of quadratic problems.

Some basic facts about the $P_G(C)$ family of problems are given below.

PROPOSITION 2.1. *For any weighted adjacency matrix C of a graph G, the number $\upsilon(G,C)$ is the optimal value of a convex quadratic problem and verifies $\alpha(G) \leq \upsilon(G,C)$, i.e., $\upsilon(G,C)$ is an upper bound on $\alpha(G)$.*

Proof. As $\lambda_{\min}(H_C) = -1$, the problem $P_G(C)$ is convex quadratic as stated. To see that $\upsilon(G,C)$ is an upper bound on $\alpha(G)$ for all matrices C, let x be a characteristic vector of any maximum independent set S of G (defined by $x_i = 1$ if $i \in S$ and $x_i = 0$ otherwise). Since the vector x is a feasible solution of $P_G(C)$ and verifies $x^T H_C\, x = 0$ (note that $x_i x_j = 0$ if $ij \in E$), we have

$$\upsilon(G,C) \geq 2e^T x - x^T x - x^T H_C\, x = 2\alpha(G) - \alpha(G) = \alpha(G),$$

i.e., $\alpha(G) \leq \upsilon(G,C)$, for all weighted adjacency matrices C of G. □

A clique of the graph $G = (V,E)$ is any subset of V such that the induced subgraph is complete. A minimum clique cover of G is a set of cliques of G that cover V with the least cardinality. This minimum number of cliques can be denoted by $\bar{\chi}(G)$ and, like the stability number, it is NP-hard to compute $\bar{\chi}(G)$. The partial graph associated with a minimum clique cover of G is a graph with the same set of vertices as that of G, and whose edges are those of the complete subgraphs induced by the cliques forming the clique cover.

PROPOSITION 2.2. *Let G be a graph with at least one edge. If M is the adjacency matrix of the partial graph associated with a minimum clique cover of G, then $\upsilon(G,M) \leq \bar{\chi}(G)$.*

Proof. Suppose that $\bar{\chi}(G) = k$ and denote by G_i, $i = 1,\dots,k$, the complete subgraphs induced by the cliques forming a minimum clique cover of G. Let x be an optimal solution of $P_G(M)$, where M is the adjacency matrix of the partial graph associated with this minimum clique cover. Note that $\lambda_{\min}(M) = -1$ since $M + I$ is formed by k all ones blocks on the diagonal (say, $J_1,\dots,J_k$), these blocks are positive semidefinite, and any J_i-block of size at least two has a zero eigenvalue. Thus

$$\upsilon(G,M) = 2e^Tx - x^T(M+I)x = \sum_{i=1}^{k} 2e_i^Tx_i - x_i^TJ_ix_i,$$

where, for each i, e_i and x_i are, respectively, the subvectors of e and x whose components correspond to the vertices of G_i. As $J_i = e_ie_i^T$ and $\left(e_i^Tx_i - 1\right)^2 \geq 0$, we have $2e_i^Tx_i - x_i^TJ_ix_i \leq 1$ for all i, hence $\upsilon(G,M) \leq k$, as required. □

Note that for any graph G with at least one edge that satisfies $\alpha(G) = \bar{\chi}(G)$ (in particular for perfect graphs), Propositions 2.1 and 2.2 allow us to define $\alpha(G)$ as follows:

$$\alpha(G) = \min_{C} \upsilon(G,C),$$

where C is a weighted adjacency matrix of G.

3. The Lovász $\vartheta(G)$ number.

The Lovász $\vartheta(G)$ number was introduced in [12] and has been subsequently studied in several publications. It is generally considered the most famous upper bound on $\alpha(G)$, for which various different formulations were established in the literature (see [9, 11]). Some of these formulations are now recalled.

An *orthonormal representation* of a graph $G = (V,E)$ with $V = \{1,2,\dots,n\}$ is a set of unit vectors $u_1,u_2,\dots,u_n$ in a Euclidean space, which are orthogonal (i.e., $u_i^Tu_j = 0$) whenever $ij \notin E$. Note that the vectors dimension is not fixed and that any graph has an orthonormal representation, considering, for example, a set of pairwise orthonormal vectors.

Lovász defined his theta number as follows:

$$\vartheta(G) = \min_{\substack{c,u_1,u_2,\dots,u_n \\ c \text{ unitary}}} \max_{i\in V} \frac{1}{\left(c^Tu_i\right)^2}, \tag{3.1}$$

where the minimum is taken over all vectors c with $||c|| = 1$ and all orthonormal representations $u_1,u_2,\dots,u_n$ of G.

As mentioned, the inequality $\alpha(G) \leq \bar{\chi}(G)$ holds true for any graph G. Both of these numbers are NP-hard to compute but they "sandwich" the number $\vartheta(G)$, which can be computed in polynomial time, as proved by Grötschel, Lovász, and Schrijver [7]. That is,

$$\alpha(G) \leq \vartheta(G) \leq \bar{\chi}(G),$$

a fact known as the *Lovász sandwich theorem* (see [11]).

The paper [12] gives several characterizations of $\vartheta(G)$. One of them is the following:

$$\vartheta(G) = \min_{A} \lambda_{\max}(A),$$

where $\lambda_{\max}(A)$ denotes the largest eigenvalue of A, and the minimum is taken over the set of all symmetric matrices $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ such that $a_{ij} = 1$ if $i = j$ or $ij \notin E$. Since we are assuming that G has at least one edge, we can eliminate the matrix ee^T from this set. In fact, if $\vartheta(G) = \lambda_{\max}(ee^T) = n$, then $\bar{\chi}(G) = n$ (recall the sandwich theorem), and thus G would have no edge.

Let A be one of the above symmetric matrices. As $A \neq ee^T$ we have that $Q = A - ee^T \neq 0$ is a weighted adjacency matrix of G. Consequently, setting $A = ee^T + Q$, $\vartheta(G)$ can be formulated as follows:

$$\vartheta(G) = \min_{Q} \lambda_{\max}(ee^T + Q), \tag{3.2}$$

where Q is a weighted adjacency matrix of G.

Another characterization of ϑ which is dual of (3.2) is the following (see [12]):

$$\vartheta(G) = \max_{B} e^T B e, \tag{3.3}$$

where $B = [b_{ij}] \in \mathbb{R}^{n\times n}$ ranges over all positive semidefinite symmetric matrices such that $b_{ij} = 0$ for $ij \in E$ and $\mathrm{Tr}(B) = 1$. ($\mathrm{Tr}(B)$ denotes the trace of B.)

4. Relating $\vartheta(G)$ and $v(G, C)$.

In this section we relate $\vartheta(G)$ with the convex quadratic upper bounds $v(G, C)$.

THEOREM 4.1. *Let G be a graph with at least one edge. Then for any weighted adjacency matrix C of graph G, we have $\vartheta(G) \leq v(G, C)$.*

Proof. Let $C = [c_{ij}]$ be a weighted adjacency matrix of $G = (V, E)$ and suppose that $P_G(C)$ is not unbounded for otherwise the theorem is true.

Let x be an optimal solution of $P_G(C)$. The Karush–Kuhn–Tucker conditions applied to this problem guarantee that the following conditions are true:

$$x \geq 0, \quad (H_C + I)\, x \geq e, \quad \text{and} \quad x^T(H_C + I)x = e^T x = v(G, C). \tag{4.1}$$

As $H_C + I$ is positive semidefinite we can write $H_C + I = U^T U$. Thus the columns of U can be thought of as an orthonormal representation of G.

Define $c = v^{-1/2} U x$, where v abbreviates $v(G, C)$. Then by (4.1), $c^T c = v^{-1} x^T \cdot (H_C + I)x = 1$ and

$$U^T c = v^{-1/2} U^T U x \geq v^{-1/2} e.$$

This inequality implies

$$\frac{1}{\left(u_i^T c\right)^2} \leq v \text{ for each } i,$$

where u_i denotes the column i of U. Recalling (3.1) we have $\vartheta(G) \leq v(G, C)$ as desired. □

This theorem asserts that $\vartheta(G)$ is not worse that any $v(G, C)$ bound. So, in particular, the inequality $\vartheta(G) \leq v(G)$ is always true. However, there are many graphs for which the value of $\upsilon(G)$ equals $\vartheta(G)$. In fact, it was proved in [4] that there is an infinite number of graphs that verify $\alpha(G) = \upsilon(G)$ and hence $\vartheta(G) = \upsilon(G)$. These graphs constitute the so-called class of graphs with convex-QP stability number. (One member of this class can be constructed by considering $L(L(G))$, where $L(G)$ is the line graph of a connected graph G with an even number of edges.)

We state now the main result of this note, which gives the announced characterization of $\vartheta(G)$ by convex quadratic programming.

Theorem 4.2. *Let G be a graph with at least one edge. If Q attains the optimum in* (3.2)*, then $\vartheta(G) = \upsilon(G, C)$, where $C = -Q$.*

Consequently, the following characterization of $\vartheta(G)$ is valid:

$$\vartheta(G) = \min_C \upsilon(G, C) = \min_C \max_{x \geq 0} \left\{ 2e^T x - x^T (H_C + I) x \right\}, \tag{4.2}$$

where C is a weighted adjacency matrix of G.

Proof. Let Q be a weighted adjacency matrix of G attaining the optimum in (3.2) and let $C = -Q$. As $\vartheta(G) = \lambda_{\max}(ee^T + Q) \geq \lambda_{\max}(Q)$, we will divide the proof of the equality $\vartheta(G) = \upsilon(G, C)$ in two cases. (To simplify the notation we will sometimes use ϑ instead of $\vartheta(G)$.)

Case 1. $\vartheta(G) = \lambda_{\max}(Q)$.

Let x attain the optimum in $P_G(C)$. Then, using the positive semidefiniteness of $I - \vartheta^{-1}(ee^T + Q)$, we have

$$\begin{aligned}
\upsilon(G, C) &= 2e^T x - x^T (H_C + I) x = 2e^T x - x^T \left(\frac{-Q}{-\lambda_{\min}(-Q)} + I \right) x \\
&= 2e^T x - x^T \left(\frac{-Q}{\lambda_{\max}(Q)} + I \right) x \\
&= 2e^T x - x^T \left(I - \vartheta^{-1} Q + \vartheta^{-1} ee^T \right) x - \vartheta^{-1} \left(e^T x \right)^2 \\
&= 2e^T x - x^T \left[I - \vartheta^{-1} \left(ee^T + Q \right) \right] x - \vartheta^{-1} \left(e^T x \right)^2 \\
&\leq 2e^T x - \vartheta^{-1} \left(e^T x \right)^2 \leq \vartheta,
\end{aligned}$$

since $\left(\vartheta^{1/2} - \vartheta^{-1/2} e^T x \right)^2 \geq 0$. So by Theorem 4.1, we have $\vartheta(G) = \upsilon(G, C)$ for this case.

Case 2. $\vartheta(G) > \lambda_{\max}(Q)$.

Let B attain the optimum in (3.3). Since $\vartheta I - ee^T - Q$ and B are positive semidefinite, we have

$$0 \leq \operatorname{Tr} \left[B(\vartheta I - ee^T - Q) \right] = \vartheta \operatorname{Tr}(B) - \operatorname{Tr}(Bee^T) - \operatorname{Tr}(BQ) = \vartheta - \vartheta - 0 = 0.$$

So $\operatorname{Tr} \left[B(\vartheta I - ee^T - Q) \right] = 0$ and then $B(ee^T + Q - \vartheta I) = 0$, i.e., the column space of B is orthogonal to the column space of $\vartheta I - ee^T - Q$. (In fact, if M and N are positive semidefinite matrices and $\operatorname{Tr}(MN) = 0$, then $MN = 0$. To see this, let $M = U^T U$ and $N = W^T W$. Then $0 = \operatorname{Tr}(MN) = \operatorname{Tr}(U^T U W^T W) = \operatorname{Tr}(W U^T U W^T)$. Since $W U^T U W^T$ is positive semidefinite, it implies that $U W^T = 0$, hence $MN = 0$.)

The inequality $\vartheta(G) > \lambda_{\max}(Q)$ implies that $\lambda_{\min}(\vartheta I - Q) > 0$ and hence $\operatorname{rank}(\vartheta I - Q) = n$. Then $\operatorname{rank}(\vartheta I - ee^T - Q) \geq n - 1$ and by the column spaces orthogonality, $\operatorname{rank}(B) \leq 1$. As $\operatorname{Tr} B = 1$, $\operatorname{rank}(B) = 1$, and then $B = \vartheta^{-1} x x^T$ for some vector x whose support is a stable set S. Since $e^T B e = \vartheta$ and $\operatorname{Tr} B = 1$, we can choose $x \geq 0$ and thus we have $e^T x = x^T x = \vartheta$. Additionally, x is a characteristic vector of S. (To see this, let y be the characteristic vector of S. Then $y^T x = e^T x = \vartheta$ and, by the Cauchy–Schwarz inequality, $(y^T x)^2 \leq (x^T x)(y^T y)$. So $\vartheta \leq |S|$ and by the maximality of ϑ, we have $y^T y = \vartheta$. Hence, the Cauchy–Schwarz inequality is satisfied with equality and this implies $x = y$.)

Using once more the orthogonality of the column spaces of B and $\vartheta I - ee^T - Q$, we conclude that $\left(ee^T + Q\right) x = \vartheta x$, and hence $-Qx = \vartheta(e - x)$. Then x satisfies the Karush–Kuhn–Tucker conditions associated with $P_G(C)$ (recall (4.1)) as

- $x \geq 0$;
- $(H_C + I)x = (\frac{-Q}{\lambda_{\max}(Q)} + I)x = \frac{-Qx}{\lambda_{\max}(Q)} + x = \frac{\vartheta}{\lambda_{\max}(Q)}(e - x) + x \geq e$, since $\vartheta \geq \lambda_{\max}(Q)$; and
- $x^T(H_C + I)x = x^T x = e^T x = \vartheta$, since x is a characteristic vector of a stable set.

Consequently, by the positive semidefiniteness of $H_C + I$, $\vartheta(G) = \upsilon(G, C)$ is also true for Case 2.

Finally, the proved equality and the definition of Q imply the characterization (4.2). $\square$

Acknowledgments. The first author thanks Domingos Cardoso and Carlos Sarrico for stimulating discussions and valuable suggestions made on a previous version of this note. He is also grateful to the UI&D Centro de Estudos de Optimização e Controlo of Aveiro University for the support given to this research.

REFERENCES

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, *Proof verification and hardness of approximation problems*, in Proceedings of the 33rd IEEE Symposium on Foundations of Computer Science, IEEE Computer Science Press, Los Alamitos, CA, 1992, pp. 14–23.

[2] C. Berge, *Graphs*, North–Holland, Amsterdam, 1991.

[3] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo, *The maximum clique problem*, in Handbook of Combinatorial Optimization, Vol. A, D. Z. Du and P. M. Pardalos, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999, pp. 1–74.

[4] D. M. Cardoso, *Convex quadratic programming approach to the maximum matching problem*, J. Global Optim., 19 (2001), pp. 291–306.

[5] D. Cvetkovic, M. Doob, and H. Sachs, *Spectra of Graphs, Theory, and Applications*, VEB Deutscher Verlag der Wissenschaften, Berlin, 1979.

[6] L. E. Gibbons, D. W. Hearn, P. M. Pardalos, and M. V. Ramana, *Continuous characterizations of the maximum clique problem*, Math. Oper. Res., 22 (1997), pp. 754–768.

[7] M. Grötschel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica, 1 (1981), pp. 169–197.

[8] M. Grötschel, L. Lovász, and A. Schrijver, *Relaxations of vertex packing*, J. Combin. Theory Ser. B, 40 (1986), pp. 330–343.

[9] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer, Berlin, 1988.

[10] C. Helmberg, *Semidefinite Programming for Combinatorial Optimization*, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, 2000.

[11] D. E. Knuth, *The sandwich theorem*, Electron. J. Combin., 1 (1994).

[12] L. Lovász, *On the Shannon capacity of a graph*, IEEE Trans. Inform. Theory, 25 (1979), pp. 1–7.

[13] L. Lovász and A. Schrijver, *Cones of matrices and set-functions and* 0-1 *optimization*, SIAM J. Optim., 1 (1991), pp. 166–190.

[14] C. J. Luz, *An upper bound on the independence number of a graph computable in polynomial time*, Oper. Res. Lett., 18 (1995), pp. 139–145.

[15] C. J. Luz and D. M. Cardoso, *A generalization of the Hoffman-Lovász upper bound on the independence number of a regular graph*, Ann. Oper. Res., 81 (1998), pp. 307–319.

[16] T. S. Motzkin and E. G. Straus, *Maxima for graphs and a new proof of a theorem of Turán*, Canad. J. Math., 17 (1965), pp. 533–540.

SIAM J. Discrete Math.
Vol. 19, No. 2, pp. 388–398

ENUMERATION OF BRANCHED COVERINGS OF NONORIENTABLE SURFACES WITH CYCLIC BRANCH POINTS*

JIN HO KWAK[†], ALEXANDER MEDNYKH[‡], AND VALERY LISKOVETS[§]

Abstract. In this paper, n-fold branched coverings of a closed nonorientable surface $\mathcal{S}$ of genus p with $r \geq 1$ cyclic branch points (that is, such that all ramification points over them are of multiplicity n) are considered. The number $N_{p,\,r}(n)$ of such coverings up to equivalence is evaluated explicitly in a closed form (without using any complicated functions such as irreducible characters of the symmetric groups). The obtained formulas depend on the parity of r and n. The method is based on some previous enumerative results and techniques for nonorientable surfaces. In particular, we generalize the approach developed for the counting of unbranched coverings of nonorientable surfaces and make use of the analytical method of roots-of-unity sums.

Key words. cyclic branch point, covering of a nonorientable surface, Ramanujan sum, von Sterneck function, fundamental group, permutation tuple, Hurwitz number

AMS subject classifications. 57M10, 20F34, 14H30

DOI. 10.1137/S0895480103424043

1. Introduction.

Throughout this paper, a surface means a compact connected 2-manifold without boundary. Recall some known concepts from algebraic topology [16]. A continuous mapping $\rho : \mathcal{T} \to \mathcal{S}$ from a surface $\mathcal{T}$ onto $\mathcal{S}$ is called a *branched covering* of multiplicity n if there exists a finite subset $B = \{b_1, \ldots, b_r\}$ of points in $\mathcal{S}$ such that the restriction of ρ on $\mathcal{T} - \rho^{-1}(B)$, $\rho|_{\mathcal{T}-\rho^{-1}(B)} : \mathcal{T} - \rho^{-1}(B) \to \mathcal{S} - B$, is an n-fold (n-sheeted) covering projection in the usual sense. The smallest subset B of $\mathcal{T}$ which has this property is called the *branch point set* of ρ. At the neighborhood of each point $x \in \rho^{-1}(B)$, the projection ρ is topologically equivalent to the complex map $z \mapsto z^m$ with some natural number m. Such an x is called a *ramification point* of ρ, and m is called the *order* of x. Denote by s^k_m the number of ramification points of order m of the mapping ρ in the preimage $\rho^{-1}(b_k)$, where $k = 1, \ldots, r$ and $m = 1, \ldots, n$. We will call the $(r \times n)$-matrix $\sigma = (s^k_m)$ the *ramification type* of the covering ρ. For any k, $(1^{s^k_1} \cdots n^{s^k_n})$ is a partition of n, that is, $\sum_m m s^k_m = n$.

Two branched coverings $\rho : \mathcal{T} \to \mathcal{S}$ and $\rho' : \mathcal{T}' \to \mathcal{S}$ are considered to be *equivalent* (or isomorphic) if there exists a homeomorphism $\eta : \mathcal{T}' \to \mathcal{T}$ such that $\rho' = \rho \circ \eta$.

The classical Hurwitz enumeration problem is to count nonequivalent n-fold coverings of $\mathcal{S}$ with a given ramification type σ. By now, only the nonorientable case for branched coverings remains open. The orientable case was, in principle, solved completely by Mednykh [18], as was the nonorientable case with unramified coverings [20]. The aim of the present work is to adjust the method of the latter article to coverings of nonorientable surfaces in the particular case when B is nonempty, $s^k_m = 0$ for $m < n$, and $s^k_n = 1$ for all $k = 1, \ldots, r$. In other words, we consider the case when

[*] Received by the editors March 10, 2003; accepted for publication (in revised form) January 6, 2005; published electronically October 31, 2005. This research was supported by Com^{2}MaC-KOSEF. http://www.siam.org/journals/sidma/19-2/42404.html

[†] Combinatorial and Computational Mathematics Center, Pohang University of Science and Technology, Pohang, 790-784, Korea (jinkwak@postech.ac.kr).

[‡] Institute of Mathematics, Novosibirsk State University, 630090, Novosibirsk, Russia (mednykh@math.nsc.ru).

[§] Institute of Mathematics, National Academy of Sciences of Belarus, 220072, Minsk, Belarus (liskov@im.bas-net.by).

every branch point is *cyclic*; i.e., it is lifted to a unique ramification point (so that the corresponding covering permutation is a full cycle of length n). Such a restriction simplifies the situation considerably, allowing elimination of irreducible characters of the symmetric groups in the formulas. Recently, this idea has been implemented successfully by two of the present authors to the cases of orientable surfaces [11]. The present work supplements this paper, extending its results to *nonorientable* surfaces. A special technique of counting the solutions of systems of linear congruences by sums of roots of unity (known also as Ramanujan's sums) is applied. See [11] for all necessary definitions, references, and additional explanations. For other useful information concerning branched coverings of nonorientable surfaces, see [8, 9, 10].

2. Preliminary results. In what follows, $\mathcal{S}$ denotes a closed nonorientable surface of genus p. The set of branch points B will be considered fixed. We denote by $\mathbf{S}_n$ the symmetric groups on n symbols and by $[g]$ the *cycle type* of a permutation $g \in \mathbf{S}_n$, that is, $[g] = (1^{s_1} \cdots n^{s_n})$ if g consists of s_m independent cycles of length m, $m = 1, \ldots, n$. For cycle types (partitions of n) we adopt the usual notational agreement to drop empty parts m^0 and to write m instead of m^1. In particular, (n) denotes the partition of n consisting of a sole part, n.

As follows from results of Hurwitz [4] and their subsequent generalizations, each covering ρ of $\mathcal{S}$ with the ramification type $\sigma = (s_m^k)$ is uniquely determined by an ordered $(p + r)$-tuple of permutations of degree n,

$$(a_1, \ldots, a_p, c_1, \ldots, c_r) \in \mathbf{S}_n^{p+r} = \underbrace{\mathbf{S}_n \times \mathbf{S}_n \times \cdots \times \mathbf{S}_n}_{p+r}, \tag{1}$$

which satisfy the relations

$$\prod_{j=1}^{p} a_j^2 \prod_{k=1}^{r} c_k = \mathbb{1}, \tag{2}$$

$$[c_k] = (1^{s_1^k} \cdots n^{s_n^k}), \quad k = 1, 2, \ldots, r, \tag{3}$$

and generate a transitive subgroup of $\mathbf{S}_n$. Here $\mathbb{1} = \mathbb{1}_n$ denotes the identity permutation. Tuples satisfying the last condition will be called *transitive.* Two coverings are *equivalent* if and only if the corresponding tuples are conjugate via a permutation from $\mathbf{S}_n$. The proof of these facts can be found, for example, in [2] or [8].

Denote by $\mathfrak{B}_{p,\,r,\,\sigma}(n)$ the set of all tuples (transitive or not) of form (1) satisfying (2) and (3), and select in $\mathfrak{B}_{p,\,r,\,\sigma}(n)$ the subset $\mathfrak{T}_{p,\,r,\,\sigma}(n)$ of transitive tuples. We set $B_{p,\,r,\,\sigma}(n) := |\mathfrak{B}_{p,\,r,\,\sigma}(n)|$ and $T_{p,\,r,\,\sigma}(n) := |\mathfrak{T}_{p,\,r,\,\sigma}(n)|$, where the vertical bars denote the cardinality of the set.

2.1. A general formula. The following result based on general formulas in terms of the irreducible characters for the number of solutions of equations in groups is valid (see [7]; cf. also [3, 6, 11, 20]).

Proposition 2.1. *The number $B_{p,\,r,\,\sigma}(n)$ of elements of the set $\mathfrak{B}_{p,\,r,\,\sigma}(n)$ is determined by the formula*

$$B_{p,\,r,\,\sigma}(n) = n! \sum_{\lambda \in D_n} \left(\prod_{k=1}^{r} \frac{\chi^{\lambda}_{s_1^k \ldots s_n^k}}{1^{s_1^k} \cdot s_1^k! \cdots n^{s_n^k} \cdot s_n^k!} \right) \left(\frac{n!}{f^{\lambda}} \right)^{p-2+r}, \tag{4}$$

where $\sigma = (s^k_m)$ is the ramification type, D_n is the set of all irreducible representations of the group $\mathbf{S}_n$, f^λ is the degree, and $\chi^\lambda_{s^k_1 \dots s^k_n} = \chi^\lambda_{[c_k]}$ is the character of permutations of the type $[c_k] = (1^{s^k_1} \cdots n^{s^k_n})$ corresponding to the representation λ.

As noted, unlike branched coverings over orientable surfaces considered in [11], no general results have been obtained so far for the number of branched coverings over nonorientable surfaces. Therefore we use here special tools sufficient for the particular case under consideration, which generalize those given in [20] and [18]. Our task is facilitated partially by a somewhat similar problem considered in [13] (see also [14]) for a class of three-dimensional manifolds.

2.2. Cyclic branch points. In the case of cyclic branch points, σ has a particular form:

$$[c_k] = (n), \quad k = 1, 2, \dots, r, \tag{5}$$

that is, all c_k are n-cycles. We denote by $T_{p,\,r}(n)$ the corresponding number of tuples, i.e., $T_{p,\,r}(n) = |\mathfrak{T}_{p,\,r,\,(n)^r}(n)| := T_{p,\,r,\,\sigma}(n)$, where σ is of form (5), i.e., $s^k_n = 1$ and $s^k_m = 0$ for $m < n$ and $k = 1, \dots, r$.

Our aim in this paper is to find the number of covering $N_{p,\,r}(n)$ up to equivalence, which coincides with the number of orbit of the symmetric group $\mathbf{S}_n$ acting by conjugation on the set $\mathfrak{T}_{p,\,r,(n)^r}(n)$. Notice also that $T_{p,\,r}(n)/(n-1)!$ is the number of subgroups of index n of the corresponding fundamental group while $N_{p,\,r}(n)$ is the number of conjugacy classes of such subgroups. In the literature, $T_{p,\,r}(n)$ are also called the corresponding *Hurwitz numbers.*

Theorem 2.2 (cf. [11]). *For any $n, r \geq 1$, and $p \geq 0$, the number of tuples* (1) *satisfying conditions* (2) *and* (5) *is the following:*

$$T_{p,\,r}(n) = \frac{(n!)^{p-1+r}}{n^r} \sum_{s=0}^{n-1} (-1)^{sr} \binom{n-1}{s}^{-(p-2+r)}. \tag{6}$$

Proof. The presence of full cycles c_k ensures transitivity, so that $\mathfrak{T}_{p,\,r,(n)^r}(n) = \mathfrak{B}_{p,\,r,(n)^r}(n)$. This simplifies the enumeration considerably; in particular by Proposition 2.1 we have

$$T_{p,\,r}(n) = n! \sum_{\lambda \in D_n} \Big(\frac{\chi^\lambda_{(n)}}{n}\Big)^r \Big(\frac{n!}{f^\lambda}\Big)^{p-2+r}. \tag{7}$$

Further we make use of the fact that characters χ^λ almost always vanish on the full cycle (n). Namely,

$$\chi^\lambda_{(n)} = \begin{cases} (-1)^s & \text{if } \lambda \vdash (1^s\, n{-}s), \quad 0 \leq s \leq n-1, \\ 0 & \text{otherwise;} \end{cases} \tag{8}$$

see, e.g., [5, Theorem 21.4] or [22, Example 7.67(a)]. Now, by the hook-length formula [22, 7.21.6] we have

$$f^\lambda = \frac{n!}{s!n(n-s-1)!} = \binom{n-1}{s} \quad \text{if} \quad \lambda \vdash (1^s n - s). \tag{9}$$

Substituting (8) and (9) into (7) we obtain (6). □

Due to formula (6), in the counting of permutation tuples for the case of cyclic branch points we have got rid of using characters.

Notice that in accordance with (6), $T_{p,\,r}(n) = 0$ for odd r and even n since in this case a product of r full cycles is an odd permutation; thus, equality (2) is impossible. However, we will not exclude this case from the subsequent consideration.

2.3. Calculations in the centralizer of a regular permutation. Since we need to count transitive permutation tuples up to conjugacy, we make use of enumerative Burnside's lemma. Accordingly we are interested in the automorphisms of tuples, that is, their centralizers. It is well known that each automorphism is a regular permutation. Hence, all permutations in such a tuple commute with this permutation h. Thus they belong to its centralizer $Z(h)$, which is of the form $\mathbb{Z}_\ell \wr \mathbf{S}_m$, where ℓ is the order of the automorphism. Now, our approach (going back to [12] and [17]) is to make necessary calculations in this wreath product so to take into account conditions (2) and (5).

Denote by b^g the action of a permutation g on an element b. Let us fix a regular permutation h of degree n and order ℓ ($\ell m = n$) which commutes with all permutations $a_1, a_2, \dots, a_p$ and $c_1, c_2, \dots, c_r$. Belonging to $Z(h) \cong \mathbb{Z}_\ell \wr \mathbf{S}_m$, they can be written in the form

$$a_i = (t^i_1, t^i_2, \dots, t^i_m; \hat{a}_i), \quad i = 1, 2, \dots, p,$$

and

$$c_k = (x^k_1, x^k_2, \dots, x^k_m; \hat{c}_k), \quad k = 1, \dots, r,$$

where all $\hat{a}_i$ and $\hat{c}_k$ belong to $\mathbf{S}_m$ and all t^i_j and x^k_j, $j = 1, \dots, m$, belong to $\mathbb{Z}_\ell$. Now using the formulas of the multiplication of permutations in $\mathbb{Z}_\ell \wr \mathbf{S}_m$ described, say, in [14], we can represent (2) as the following system of congruences:

(10)
$$t^1_j + t^1_{j^{\hat{a}_1}} + t^2_{j^{\hat{a}^2_1}} + \dots + t^p_{j^{\hat{a}^2_1 \cdots \hat{a}^2_{p-1}}} + t^p_{j^{\hat{a}^2_1 \cdots \hat{a}^2_{p-1} \hat{a}_p}} + x^1_{j^{\hat{a}^2_1 \cdots \hat{a}^2_p}} + x^2_{j^{\hat{a}^2_1 \cdots \hat{a}^2_p \hat{c}_1}} + \dots \equiv 0 \pmod{\ell}$$

for $j = 1, \dots, m$ together with the equation

$$\hat{a}^2_1 \hat{a}^2_2 \cdots \hat{a}^2_p \hat{c}_1 \cdots \hat{c}_r = \mathbb{1}_m. \tag{11}$$

It is easy to see (see [18]) that in these terms, condition (5) is expressed as follows:

$$(x^k_1 + x^k_2 + \dots + x^k_m, \ell) = 1, \quad k = 1, 2, \dots, r, \quad r \geq 1, \tag{12}$$

where again (,) denotes the greatest common divisor, and

$$[\hat{c}_k] = (m), \quad k = 1, 2, \dots, r. \tag{13}$$

Now we are interested in the number of solutions of the system of (10) and (12). This number proves (as we will see later) to be independent of a specific choice of the tuple $(a_1, \dots, a_p, c_1, \dots, c_r)$. More generally, let us consider an *arbitrary* $(2p + r)$-tuple of permutations of degree m, $(\alpha_1, \beta_1, \dots, \alpha_p, \beta_p, \gamma_1, \dots, \gamma_r)$, where $\gamma_1, \dots, \gamma_r$ are full cycles, and let $M = M_{p,\,r,\,m}(\ell)$ denote the number of solutions of the system (12) and (14) in $\mathbb{Z}_\ell$, where

$$t^1_{j^{\alpha_1}} + t^1_{j^{\beta_1}} + \dots + t^p_{j^{\alpha_p}} + t^p_{j^{\beta_p}} + x^1_{j^{\gamma_1}} + \dots + x^r_{j^{\gamma_r}} \equiv 0 \pmod{\ell}, \quad j = 1, \dots, m. \tag{14}$$

The following lemma is a crucial technical result of this work.

LEMMA 2.3. *For any tuple* $(\alpha_1, \beta_1, \dots, \alpha_p, \beta_p, \gamma_1, \dots, \gamma_r) \in \mathbf{S}_m^{2p+r}$, *where all* γ_i *are full cycles, the number of solutions of the system of* (12) *and* (14) *in* $\mathbb{Z}_\ell$ *is determined by the following formula:*

$$M_{p,\,r,\,m}(\ell) = \ell^{m(p-1+r)-r}\phi(\ell)^r \times \begin{cases} 1 & \textit{for } \ell \textit{ odd}, \\ 2 & \textit{for } \ell \textit{ even},\ r \textit{ even}, \\ 0 & \textit{for } \ell \textit{ even},\ r \textit{ odd}, \end{cases} \tag{15}$$

where $\phi(\ell)$ *is the Euler function.*

Proof. Generally we make use of the same technique as in the appendix of [13] (see also [20]). Denoting by f_j the left-hand-side expressions of (14), we introduce the following polynomials of $z_1, \dots, z_m$:

$$P(z_1, \dots, z_m) := \sum_{\substack{\forall i,j,k \ \ 1 \le t_j^i, x_j^k \le \ell \\ (x_1^k + \cdots + x_m^k, \ell) = 1}} \prod_{j=1}^m z_j^{f_j}. \tag{16}$$

Then the number of solutions of the system (12) and (14) modulo ℓ coincides with the sum of the coefficients of $P(z_1, \dots, z_m)$, all indices of which are divisible by ℓ, and, consequently, is given by the formula

$$M = \frac{1}{\ell^m} \sum_{\substack{1 \le \ell_1 \le \ell \\ \cdots \\ 1 \le \ell_m \le \ell}} P(\varepsilon^{\ell_1}, \dots, \varepsilon^{\ell_m}), \tag{17}$$

where $\varepsilon = \sqrt[\ell]{1} = \exp \frac{2\pi i}{\ell},\ i = \sqrt{-1}$.

Changing the order of the factors in (16) by applying $\alpha_i^{-1}, \beta_i^{-1}$, and γ_i^{-1} to subscripts, one can represent $\prod_{j=1}^m z_j^{f_j}$ as follows:

$$P(z_1, \dots, z_m) = \sum_{\substack{1 \le t_j^i, x_j^k \le \ell \\ (x_1^k + \cdots + x_m^k, \ell) = 1}} \prod_{j=1}^m \Big(z_{j^{\alpha_1^{-1}}}^{t_j^1} z_{j^{\beta_1^{-1}}}^{t_j^1} \cdots z_{j^{\alpha_p^{-1}}}^{t_j^p} z_{j^{\beta_p^{-1}}}^{t_j^p} z_{j^{\gamma_1^{-1}}}^{x_j^1} \cdots z_{j^{\gamma_r^{-1}}}^{x_j^r} \Big),$$

whence by elementary, although tedious, transformations,

$$\begin{aligned} & P(\varepsilon^{\ell_1}, \dots, \varepsilon^{\ell_m}) \\ = \ & \sum_{\substack{1 \le t_j^i, x_j^k \le \ell \\ (x_1^k + \cdots + x_m^k, \ell) = 1}} \prod_{j=1}^m \varepsilon^{t_j^1(\ell_{j^{\alpha_1^{-1}}} + \ell_{j^{\beta_1^{-1}}})} \cdots \varepsilon^{t_j^p(\ell_{j^{\alpha_p^{-1}}} + \ell_{j^{\beta_p^{-1}}})} \varepsilon^{x_j^1 \ell_{j^{\gamma_1^{-1}}}} \cdots \varepsilon^{x_j^r \ell_{j^{\gamma_r^{-1}}}} \\ = \ & \prod_{j=1}^m \left(\sum_{t_j^1 = 1}^{\ell} \varepsilon^{t_j^1(\ell_{j^{\alpha_1^{-1}}} + \ell_{j^{\beta_1^{-1}}})} \cdots \sum_{t_j^p = 1}^{\ell} \varepsilon^{t_j^p(\ell_{j^{\alpha_p^{-1}}} + \ell_{j^{\beta_p^{-1}}})} \right) \prod_{k=1}^r \sum_{\substack{1 \le x_1^k, \dots, x_m^k \le \ell \\ (x_1^k + \cdots + x_m^k, \ell) = 1}} \varepsilon^{x_j^k \ell_{j^{\gamma_k^{-1}}}} \\ = \ & \prod_{j=1}^m [\ell\delta(0, \ell_{j^{\alpha_1^{-1}}} + \ell_{j^{\beta_1^{-1}}}) \cdots \ell\delta(0, \ell_{j^{\alpha_p^{-1}}} + \ell_{j^{\beta_p^{-1}}})] \delta(\ell_1, \ell_2, \dots, \ell_m) \Phi(\ell_1, \ell)^r \ell^{r(m-1)}. \end{aligned} \tag{18}$$

Here we use a multivariable δ-function defined as follows:

$$\delta(a,b,c,\dots) := \begin{cases} 1 & \text{if } a \equiv b \equiv c \equiv \dots \pmod{\ell}, \\ 0 & \text{otherwise.} \end{cases}$$

The last equality in (18) is based on the following claim.

Claim 1. *$\Sigma := \sum_{\substack{1 \le x_1,\dots,x_m \le \ell \\ (x_1+\dots+x_m,\ell)=1}} \varepsilon^{x_j \ell_j} = \delta(\ell_1,\dots,\ell_m)\ell^{m-1}\Phi(\ell_1,\ell)$, where $\Phi(u,\ell)$ is the von Sterneck function (known also as Ramanujan's sum): the sum of the primitive ℓth roots of unity in the power u:*

$$\Phi(u,\ell) := \sum_{x:\ (x,\ell)=1} \varepsilon^{xu}.$$

In turn, Claim 1 relies on the following well-known identity.

Claim 2. *For any two integers a and b, we have $\sum_{x=1}^{\ell} \varepsilon^{x(a-b)} = \ell\delta(a,b)$.* □

We have

$$\Sigma = \sum_{x_1=1}^{\ell} \varepsilon^{(\ell_1-\ell_m)x_1} \cdots \sum_{x_{m-1}=1}^{\ell} \varepsilon^{(\ell_{m-1}-\ell_m)x_{m-1}} \sum_{(x,\ell)=1} \varepsilon^{\ell_m x} = \delta(\ell_1,\dots,\ell_m)\ell^{m-1}\Phi(\ell_m,\ell),$$

where $x := x_1 + \dots + x_m$. Besides, $\delta(\ell_1,\dots,\ell_m)\Phi(\ell_m,\ell) = \delta(\ell_1,\dots,\ell_m)\Phi(\ell_1,\ell)$ since both products vanish unless $\ell_1 = \ell_2 = \dots = \ell_m$. These arguments prove Claim 1. □

Return to the proof of Lemma 2.3. The factor $\delta(\ell_1,\dots,\ell_m)$ in the last expression in (18) shows that the polynomial $P(\varepsilon^{\ell_1},\dots,\varepsilon^{\ell_m})$ does not vanish only if all ℓ_j coincide,

$$\ell_1 = \dots = \ell_m = \lambda, \tag{19}$$

in which case $P(\varepsilon^{\ell_1},\dots,\varepsilon^{\ell_m}) = \ell^{mp+(m-1)r}\Phi(\lambda,\ell)^r\Delta$, where

$$\Delta = \prod_{j=1}^{m} [\delta(\ell_{j^{\alpha_1^{-1}}}, -\ell_{j^{\beta_1^{-1}}}) \cdots \delta(\ell_{j^{\alpha_p^{-1}}}, -\ell_{j^{\beta_p^{-1}}})]. \tag{20}$$

Thus, Δ is always equal to 0 or 1. Now it is clear that (regardless of α_i, β_i) in view of (19), Δ does not vanish if and only if $\lambda \equiv -\lambda \pmod{\ell}$ or, equivalently,

$$2\lambda \equiv 0 \pmod{\ell}. \tag{21}$$

This equation has only the trivial solution $\lambda \equiv 0 \pmod{\ell}$ if ℓ is odd, and it has the additional solution $\lambda \equiv \ell/2 \pmod{\ell}$ if ℓ is even.

We conclude that

$$P(\varepsilon^{\ell_1},\dots,\varepsilon^{\ell_m}) = \ell^{mp+(m-1)r}\Phi(0,\ell)^r$$

if $\ell_1 = \dots = \ell_m = 0$,

$$P(\varepsilon^{\ell_1},\dots,\varepsilon^{\ell_m}) = \ell^{mp+(m-1)r}\Phi(\ell/2,\ell)^r$$

if ℓ is even, and $\ell_1 = \dots = \ell_m = \ell/2$, and

$$P(\varepsilon^{\ell_1},\dots,\varepsilon^{\ell_m}) = 0$$

in all other cases.

As was shown by Hölder,

$$\Phi(x,n) = \frac{\phi(n)}{\phi(\frac{n}{(x,n)})}\,\mu\left(\frac{n}{(x,n)}\right), \tag{22}$$

where $\mu(n)$ is the number-theoretic Möbius function [1, p. 164] (cf. [21]). It follows that $\Phi(0,\ell) = \phi(\ell)$ and $\Phi(\ell/2,\ell) = -\phi(\ell)$.

Substitute these values into the above expressions for $P(\varepsilon^{\ell_1},\dots,\varepsilon^{\ell_m})$ and substitute them into (17). Taking into account that $\phi(\ell)^r + (-\phi(\ell))^r = 0$ if r is odd and $\phi(\ell)^r + (-\phi(\ell))^r = 2\phi(\ell)^r$ if r is even, we finally obtain (15). □

3. Enumeration. The main result of this paper is the following.

THEOREM 3.1. *The number $N_{p,\,r}(n)$ of nonequivalent n-fold coverings of a closed nonorientable surface of genus p with $r \geq 1$ cyclic branch points is expressed by the following formulas:*

$$N_{p,\,r}(n) = \begin{cases} n^{p-2} \displaystyle\sum_{\substack{\ell|n\\ \ell m=n}} \ell^{(m-1)\nu}\phi(\ell)^r(2,\ell)\sum_{s=0}^{m-1}[s!(m-s-1)!]^\nu & \textit{for } r \textit{ even},\\ n^{p-2} \displaystyle\sum_{\substack{\ell|n\\ \ell m=n}} \ell^{(m-1)\nu}\phi(\ell)^r\sum_{s=0}^{m-1}(-1)^s[s!(m-s-1)!]^\nu & \textit{for } r \textit{ odd},\ n \textit{ odd},\\ 0 & \textit{for } r \textit{ odd},\ n \textit{ even}, \end{cases} \tag{23}$$

where $\nu := p-2+r$ is the characteristic of $\mathcal{S} - B$, $\phi(\ell)$ is the Euler function, and $(2,\ell)$ denotes the greatest common divisor of the numbers 2 and ℓ.

Proof. Recall that the number of coverings $N_{p,\,r}(n)$ coincides with the number of orbits of the symmetric group $\mathbf{S}_n$ acting by conjugation on the set $\mathfrak{T}_{p,\,r,(n)^r}(n)$. By applying Burnside's lemma we obtain

$$N_{p,\,r}(n) = \frac{1}{n!}\sum_{\substack{\ell|n\\ \ell\, m=n}} \frac{n!}{m!\ell^m}\widetilde{T}_{p,\,r}(\ell^m), \tag{24}$$

where $\widetilde{T}_{p,\,r}(\ell^m)$ denotes the number of tuples (1) satisfying (2) and (5) and commuting with a fixed regular permutation h of order ℓ. As we saw, these are permutation tuples satisfying (in terms of the centralizer $Z(h)$) conditions (10)–(13). Since restrictions (10) and (12) are *independent* of (11) and (13), multiplying the numbers of solutions of both problems, we obtain in the designations adopted above the following proposition.

PROPOSITION 3.2.

$$\widetilde{T}_{p,\,r}(\ell^m) = T_{p,\,r}(m)M_{p,\,r,\,m}(\ell). \tag{25}$$

Now we make use of formulas (6) and (15). Notice that the last factor in (15) can be represented equivalently as follows:

$$\begin{cases} (2,\ell) & \text{for } r \text{ even},\\ \quad 1 & \text{for } r \text{ odd},\ \ell \text{ odd},\\ \quad 0 & \text{for } r \text{ odd},\ \ell \text{ even}. \end{cases} \tag{26}$$

Substituting expressions (25), (6), and (15) (taking into account (26)) into (24), after elementary transformations we obtain the first two formulas (23). Now consider the last case, when r is odd and n is even. According to (15) (or (26)), for even ℓ dividing n, the factor $M_{p,\,r,\,m}(\ell) = 0$. Now suppose that ℓ is odd. Then m is even. In this case, $\sum_{s=0}^{m-1}(-1)^s\binom{m-1}{s}^{-\nu} = 0$ since $\binom{m-1}{s}^{-\nu} = \binom{m-1}{m-1-s}^{-\nu}$ and s and $m-1-s$ are of different parity. Thus, $T_{p,r}(m) = 0$ and $N_{p,\,r}(n) = 0$. □

Remark 1. In our case, the covering surface is *nonorientable*. Indeed, since the permutation c_1 is a full cycle, for any permutation $a_1 \in \mathbf{S}_n$ there exists an integer k such that the permutation $a_1c_1^k$ fixes the element 1. The word $a_1c_1^k$ contains an odd number of letters a_j; therefore, by the familiar criterion [2], this means that the corresponding covering surface is nonorientable. Besides, by the Riemann–Hurwitz formula it is of characteristic $n\nu$.

Remark 2. It is interesting to compare (23) with the formula for the number $N^{\mathrm{o}}_{g,\,r}(n)$ of the corresponding coverings of an *orientable* surface of genus g. According to [11] (in a slightly modified form),

$$N^{\mathrm{o}}_{g,\,r}(n) = n^{2g-2} \sum_{\substack{\ell\mid n\\ \ell m=n}} \ell^{(m-1)\nu}\psi(r,\ell) \sum_{s=0}^{m-1} (-1)^{sr}[s!(m-s-1)!]^{\nu}, \tag{27}$$

where $\psi(r,\ell) := \sum_{k=1}^{\ell} \Phi(k,\ell)^r$ and $\nu := 2g-2+r$. At the same time, for the number of the corresponding permutation tuples $T^{\mathrm{o}}_{g,\,r}(n)$ we conclude from [11] and formula (6) above for $p = 2g$ that

$$T^{\mathrm{o}}_{g,\,r}(n) = T_{2g,\,r}(n). \tag{28}$$

Now let us express $N_{p,\,r}(n)$ in terms of $T_{p,\,r}(m)$, $m|n$. Formula (6) can be rewritten in the following form:

$$T_{p,\,r}(n) = n!n^{p-2}\sum_{s=0}^{n-1}(-1)^{sr}[s!(n-s-1)!]^{\nu}. \tag{29}$$

In (23) we can join the first two formulas with the help of the greatest common divisor of three numbers $(2,\ell,r)$. After that, substituting there the right-hand-side expression of (29), we obtain

$$N_{p,\,r}(n) = \begin{cases} 0 & \text{for } r \text{ odd, } n \text{ even,} \\ n^{p-2} \displaystyle\sum_{\substack{\ell\mid n\\ \ell m=n}} \frac{\ell^{(m-1)\nu}\phi(\ell)^r(2,\ell,r)}{m^{p-1}} \, \frac{T_{p,\,r}(m)}{(m-1)!} & \text{otherwise.} \end{cases} \tag{30}$$

For comparison, formula (27) can be rewritten in a similar form as follows:

$$N^{\mathrm{o}}_{g,\,r}(n) = n^{2g-2} \sum_{\substack{\ell\mid n\\ \ell m=n}} \frac{\ell^{(m-1)\nu}\psi(r,\ell)}{m^{2g-1}} \, \frac{T^{\mathrm{o}}_{g,\,r}(m)}{(m-1)!}. \tag{31}$$

Here are the values of $N_{p,\,r}(n)$ for $n \le 7$. For even r,

$N_{p,\,r}(1) = 1,$
$N_{p,\,r}(2) = 2^p,$
$N_{p,\,r}(3) = 3^{p-2}(2^{\nu+1} + 1 + 2^r),$

$N_{p,\,r}(4) = 2 \cdot 4^{p-2}(6^\nu + 3 \cdot 2^\nu + 2^r),$
$N_{p,\,r}(5) = 5^{p-2}(2 \cdot 24^\nu + 2 \cdot 6^\nu + 4^\nu + 4^r),$
$N_{p,\,r}(6) = 2 \cdot 6^{p-2}(120^\nu + 24^\nu + 12^\nu + 3^\nu \cdot 2^r + 2 \cdot 8^\nu + 4^\nu + 2^r),$
$N_{p,\,r}(7) = 7^{p-2}(2 \cdot 720^\nu + 2 \cdot 120^\nu + 2 \cdot 48^\nu + 36^\nu + 6^r),$

and for odd r and odd n, $N_{p,\,r}(1) = 1$,

$N_{p,\,r}(3) = 3^{p-2}(2^{\nu+1} - 1 + 2^r),$
$N_{p,\,r}(5) = 5^{p-2}(2 \cdot 24^\nu - 2 \cdot 6^\nu + 4^\nu + 4^r),$
$N_{p,\,r}(7) = 7^{p-2}(2 \cdot 720^\nu - 2 \cdot 120^\nu + 2 \cdot 48^\nu - 36^\nu + 6^r).$

3.1. Coverings of the projective plane and the Klein bottle. Consider now the particular cases when $\nu = 1$. These are coverings of the projective plane and the Klein bottle with two and one branch points, respectively.

COROLLARY 3.3. *The number of nonequivalent n-fold coverings of the projective plane with two cyclic branch points is given by the formula*

$$N_{1,\,2}(n) = \frac{1}{n} \sum_{\substack{\ell \mid n \\ \ell m = n}} (2, \ell)\phi(\ell)^2 \ell^{m-1} \sum_{s=0}^{m-1} s!(m-s-1)!. \tag{32}$$

In particular, if $n = q$ is an odd prime, then

$$N_{1,\,2}(q) = \frac{1}{q}\left((q-1)^2 + \sum_{s=0}^{q-1} s!(q-s-1)!\right). \qquad \square$$

The numerical values for $n = 1,2,3,4,5,6,7,8$ are $1,2,3,8,16,64,264,1580$.

Formula (32) can be slightly simplified due to the following familiar identity [23] (see also [15] and references therein):

$$\sum_{s=0}^{n} s!(n-s)! = \frac{(n+1)!}{2^n} \sum_{j=0}^{n} \frac{2^j}{j+1}. \tag{33}$$

COROLLARY 3.4. *The number of nonequivalent n-fold coverings of the Klein bottle with one cyclic branch point is given by the formula*

$$N_{2,\,1}(n) = 2 \sum_{\substack{\ell \mid n \\ \ell m = n}} \frac{m!\ell^{m-1}\phi(\ell)}{m+1} \tag{34}$$

if n is odd and $N_{2,\,1}(n) = 0$ if n is even. In particular, if $n = q$ is an odd prime, then $N_{2,\,1}(q) = q - 1 + 2q!/(q+1)$.

Proof. $N_{2,\,1}(n)$ vanishes for even n, and for odd n we should take the second formula of (23) with $\nu = 1$. Now for odd m, the following elementary identity is valid [19] (see also [22, Example 7.67(c)]):

$$\sum_{s=0}^{m-1} (-1)^s [s!(m-s-1)!] = \frac{2m!}{m+1}; \tag{35}$$

the corollary follows. $\square$

The numerical values for $n = 1,3,5,7,9,11$ are $1,5,44,1266,72636,6652810$.

Other numerical data for the projective plane and the Klein bottle are contained in Tables 3.1 and 3.2.

TABLE 3.1
The number of n-sheeted coverings of the projective plane $(p = 1)$ with r cyclic branch points.

$n\backslash r$	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	0	2	0	2	0	2	0
3	1	3	5	11	21	43	85
4	0	8	0	128	0	3968	0
5	1	16	232	5680	132448	3189184	76426624
6	0	64	0	581696	0	8297164544	0
7	1	264	144504	107174448	76724477856	55290551845824	39803169903525504

TABLE 3.2
The number of n-sheeted coverings of the Klein bottle $(p = 2)$ with r cyclic branch points.

$n\backslash r$	1	2	3	4	5	6
1	1	1	1	1	1	1
2	0	4	0	4	0	4
3	5	13	23	49	95	193
4	0	104	0	2720	0	93824
5	44	1256	27344	666656	15911744	382307456
6	0	30608	0	415444544	0	5972357328128
7	1266	1071540	743214744	537904137744	386934209149536	278634137614009920

3.2. Asymptotics. It is evident that in formula (23) for fixed p and r and growing n, the term with $\ell = 1$ dominates (of course, unless $\nu = 0$ or r is odd and n is even). In turn, the dominating terms of its internal sum for $\ell = 1$ correspond to $s = 0$ and $s = n - 1$ and are equal to $(n-1)!^{\nu}$. Therefore we have the next corollary.

COROLLARY 3.5. *Asymptotically for fixed p and r (except for $p = r = 1$),*

$$N_{p,\,r}(n) \sim 2\frac{n!^{\nu}}{n^{r}} = 2n!^{p-2}(n-1)!^{r} \tag{36}$$

as $n \to \infty$, where n is odd if r is odd. □

By (6), $T_{p,\,r}(n) \sim 2n!^{p-1}(n-1)!^{r}$ as $n \to \infty$, with the same restrictions. So

$$N_{p,\,r}(n) \sim \frac{T_{p,\,r}(n)}{n!}.$$

Notice that $n!^{p}(n-1)!^{r}$ is the number of tuples (1) satisfying (5). As we see, (2) diminishes this number asymptotically $n!/2$ times.

Acknowledgment. The second and third authors are thankful to Com^{2}MaC of the Pohang University of Science and Technology for support and hospitality during their visit to Pohang, when the results of the present paper were obtained.

REFERENCES

[1] T. M. APOSTOL, *Introduction to Analytic Number Theory*, Springer, New York, 1976.

[2] C. L. EZELL, *Branch point structure of covering maps onto nonorientable surfaces*, Trans. Amer. Math. Soc., 243 (1978), pp. 123–133.

[3] G. FROBENIUS AND I. SCHUR, *Über die reelen Darstellung der endlichen Gruppen*, Sitzer. Königlich Preuss. Akad. Wiss. Berlin, 1906, pp. 186–208.

[4] A. HURWITZ, *Über Riemann'sche Flächen mit gegebenen Verzweigungspunkten*, Math. Ann., 39 (1891), pp. 1–60.

[5] G. D. JAMES, *The Representation Theory of the Symmetric Group*, Lecture Notes in Math. 682, Springer, Berlin, 1978.

[6] G. A. Jones, *Enumeration of homomorphisms and surface-coverings*, Quart. J. Math. Oxford, 46 (1995), pp. 485–507.

[7] A. Kerber and B. Wagner, *Gleichungen in endlichen Gruppen*, Arch. Math. (Basel), 35 (1980), pp. 252–262.

[8] J. H. Kwak and J. Lee, *Enumeration of graph coverings, surface branched coverings and related group theory*, in Combinatorial Computational Mathematics: Present and Future, S. Hong, J. H. Kwak, et al., eds., Word Scientific, Singapore, 2001, pp. 97–161.

[9] J. H. Kwak, J. Lee, and A. D. Mednykh, *Enumerating branched surface coverings from unbranched ones*, LMS J. Comput. Math., 6 (2003), pp. 89–104.

[10] J. H. Kwak, J. Lee, and Y. Shin, *Balanced regular coverings of a signed graph and regular branched orientable surface coverings over a nonorientable surface*, Discrete Math., 275 (2004), pp. 177–193.

[11] J. H. Kwak and A. Mednykh, *Enumeration of branched coverings of closed surfaces whose branched orders coincide with multiplicity*, Studia Sci. Math. Hungarica, to appear; also available online from http://com2mac.postech.ac.kr/papers/2002/02-05.ps.

[12] V. A. Liskovets, *Towards the enumeration of subgroups of the free group*, Dokl. Akad. Nauk BSSR, 15 (1971), pp. 6–9 (in Russian).

[13] V. Liskovets and A. Mednykh, *On the Number of Subgroups in the Fundamental Groups for a Class of Seifert Fibre Spaces*, Preprint MATH-AL-15-1997, TU Dresden, Dresden, Germany, 1997.

[14] V. Liskovets and A. Mednykh, *The number of subgroups in the fundamental groups of some non-orientable* 3*-manifolds*, in Formal Power Series and Algebraic Combinatorics (Moscow, 2000), D. Krob et al., eds., Springer, Berlin, 2000, pp. 276–287.

[15] T. Mansour, *Combinatorial identities and inverse binomial coefficients*, Adv. in Appl. Math., 28 (2002), pp. 196–202.

[16] W. S. Massey, *Algebraic Topology: An Introduction*, Grad. Texts in Math. 56, Springer, New York, 1977.

[17] A. D. Mednykh, *On the Hurwitz problem on the number of nonequivalent coverings over a compact Riemann surface*, Sibirsk. Mat. Zh., 23 (1982), pp. 155–160, 222 (in Russian); Siber. Math. J., 23 (1983), pp. 415–420 (in English).

[18] A. D. Mednykh, *Nonequivalent coverings of Riemann surfaces with a prescribed ramification type*, Sibirsk. Mat. Zh., 25 (1984), pp. 120–142 (in Russian); Siber. Math. J., 25 (1984), pp. 606–625 (in English).

[19] A. D. Mednykh, *Branched coverings of Riemann surfaces whose branch orders coincide with the multiplicity*, Commun. Algebra, 18 (1990), pp. 1517–1533.

[20] A. D. Mednykh and G. G. Pozdnyakova, *Number of nonequivalent coverings over a compact nonorientable surface*, Sibirsk. Mat. Zh., 27 (1986), pp. 123–131 (in Russian); Siber. Math. J., 27 (1986), pp. 99–106 (in English).

[21] C. A. Nicol and H. S. Vandiver, *A von Sterneck arithmetical function and restricted partitions with respect to modulus*, Proc. Natl. Acad. Sci. USA, 40 (1954), pp. 825–835.

[22] R. P. Stanley, *Enumerative Combinatorics*, Vol. 2, Cambridge University Press, Cambridge, UK, 1999.

[23] B. Sury, *Sum of the reciprocals of the binomial coefficients*, European J. Combin., 14 (1993), pp. 351–353.

SIAM J. DISCRETE MATH.
Vol. 19, No. 2, pp. 399–419

NONSEPARATING PLANAR CHAINS IN 4-CONNECTED GRAPHS*

SEAN CURRAN†, ORLANDO LEE‡, AND XINGXING YU§

Abstract. In this paper, we describe an $O(|V(G)||E(G)|)$ algorithm for finding a nonseparating planar chain in a 4-connected graph G, which will be used to decompose an arbitrary 4-connected graph into planar chains. This work was motivated by the study of a multitree approach to reliability in distributed networks, as well as the study of nonseparating induced paths in highly connected graphs.

Key words. connectivity, planar chain, nonseparating, algorithm

AMS subject classifications. 05C40, 05C85, 05C38, 05C75

DOI. 10.1137/S0895480103434580

1. Introduction.

Let $G = (V(G), E(G))$ denote a graph with *vertex set* $V(G)$ and *edge set* $E(G)$. We use the notation xy (or yx) to represent an edge with ends x and y. For any $S \subseteq V(G)$, let $G[S]$ denote the subgraph of G with $V(G[S]) = S$ and $E(G[S])$ consisting of the edges of G with both ends in S; we say that $G[S]$ is the subgraph of G *induced* by S. Let $G - S$ denote $G[V(G) - S]$. A subgraph H of G is an *induced subgraph* of G if $G[V(H)] = H$. We also say that H is *induced* in G. A graph G is *k-connected*, where k is a positive integer, if $|V(G)| \geq k + 1$ and, for any $S \subset V(G)$ with $|S| \leq k - 1$, $G - S$ is connected. A subgraph H of G is *nonseparating* if $G - V(H)$ is connected.

In 1984, Itai and Rodeh [10] proposed a multitree approach to reliability in distributed networks. Let G be a graph and $r \in V(G)$. We may view G as a distributed network with a root r and the vertices of G as processors. A fault-tolerant communication scheme can be designed for this network if we are able to find spanning trees of G which are independent [6, 10]. For a tree T and $x, y \in V(T)$, let $T[x, y]$ denote the unique path from x to y in T. A *rooted tree* T is a tree with a specified vertex called the *root* of T. Let T and T' be trees in a graph rooted at r. We say that T and T' are *independent* if for each vertex $x \in V(T) \cap V(T')$, the paths $T[r, x]$ and $T'[r, x]$ have no vertex in common except for r and x.

Itai and Rodeh [10] developed a linear time algorithm that given any vertex r in a 2-connected graph G finds two independent spanning trees of G rooted at r. Later, Cheriyan and Maheshwari [3] proved that for any vertex r in a 3-connected graph G, there exist three independent spanning trees of G rooted at r. Furthermore, they gave an $O(|V(G)|^2)$ algorithm for finding these trees. Itai and Zehavi [11] proved

*Received by the editors September 17, 2003; accepted for publication (in revised form) January 4, 2005; published electronically October 31, 2005.

http://www.siam.org/journals/sidma/19-2/43458.html

†School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332. The work of this author was partially supported by an NSF VIGRE grant.

‡Instituto de Computação, Universidade Estadual de Campinas, Avenida Albert Einstein, 1251, Caixa Postal 6176, 13083–970 Campinas, SP, Brazil. The work of this author was supported by CNPq (Proc: 200611/00-3), Brazil.

§School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332, and Center for Combinatorics, LPMC, Nankai University, Tianjin, 300071, People's Republic of China (yu@math.gatech.edu). The work of this author was partially supported by NSF grant DMS 9970527 and NSA grant MDA 904-03-1-0052.

independently that every 3-connected graph contains three independent spanning trees (rooted at any vertex), and they conjectured the following.

Conjecture 1.1. *Let G be a k-connected graph and let $r \in V(G)$. Then there exist k independent spanning trees of G rooted at r.*

A *contractible* edge in a k-connected graph is an edge whose contraction results in a new k-connected graph. Itai and Zehavi's proof for the 3-connected case relies on the existence of a contractible edge. On the other hand, for every $k \geq 4$ there exist infinitely many k-connected graphs with no contractible edges. In view of this fact, it would be interesting to know if Conjecture 1.1 holds for $k = 4$. The 4-connected case of Conjecture 1.1 is also important in terms of applications, since four independent spanning trees ensure at a reasonable cost a higher degree of reliability in distributed networks. Huck [8] proved Conjecture 1.1 for planar 4-connected graphs. Miura et al. [14] gave a linear algorithm for finding four independent rooted spanning trees in a planar 4-connected graph.

Itai and Rodeh's algorithm [10] for constructing two independent spanning trees relies on "ear decompositions" of graphs. Cheriyan and Maheshwari [3] used the concept of nonseparating ear decomposition to construct three independent spanning trees in 3-connected graphs. The first step in their approach is to find a nonseparating cycle which avoids a given vertex. A cycle C *avoids* a vertex v if $v \notin V(C)$.

Theorem 1.2. *Let G be a 3-connected graph, let $e \in E(G)$, and let $u \in V(G)$ be nonincident to e. Then G has a nonseparating induced cycle through e and avoiding u. Moreover, such a cycle can be found in $O(|V(G)| + |E(G)|)$ time.*

The existence of a nonseparating induced cycle in Theorem 1.2 was proved by Tutte [21], and the algorithmic part was done by Cheriyan and Maheshwari [3, Theorem 5]. In general, it is not true that given an edge e in a 4-connected graph G, there exists a cycle C through e such that $G - V(C)$ is 2-connected. In this paper we are concerned with the problem of finding a nonseparating planar chain in a 4-connected graph whose deletion results in a 2-connected graph. (A nonseparating planar chain may be viewed as a generalization of the concept of a nonseparating path.) We give an efficient algorithm for solving this problem. Our result has some interesting consequences (section 4) and will be used in a forthcoming paper to decompose an arbitrary 4-connected graph into planar chains.

To describe precisely our result, we need to introduce the concept of chain and planar chain. A *block* of a graph G is either a maximal 2-connected subgraph of G or a subgraph of G induced by a cut edge. A block is *nontrivial* if it is 2-connected, and it is *trivial* otherwise.

Definition 1.3. *A connected graph H is a* chain *if its blocks can be labeled as $B_1, \dots, B_k$, where $k \geq 1$ is an integer, and its cut vertices can be labeled as $v_1, \dots, v_{k-1}$ such that*

(i) $V(B_i) \cap V(B_{i+1}) = \{v_i\}$ *for* $1 \leq i \leq k-1$ *and*

(ii) $V(B_i) \cap V(B_j) = \emptyset$ *if* $|i - j| \geq 2$ *and* $1 \leq i, j \leq k$.

We let $H := B_1 v_1 B_2 v_2 \dots v_{k-1} B_k$ denote this situation. If $k \geq 2$, $v_0 \in V(B_1) - \{v_1\}$ and $v_k \in V(B_k) - \{v_{k-1}\}$, or, if $k = 1$, $v_0, v_k \in V(B_1)$ and $v_0 \neq v_k$, then we say that H is a v_0-v_k chain*, and we denote this by $H := v_0 B_1 v_1 \dots v_{k-1} B_k v_k$. We usually fix v_0 and v_k, and we refer to them as the* ends *of H. See Figure* 1 *for an example with $k = 5$.*

A *plane graph* is a graph which is drawn in the plane with no pair of edges crossing. Let G be a graph with distinct vertices a, b, c, and d. We say that the quintuple (G, a, b, c, d) is *planar* if G can be drawn in a closed disc in the plane with

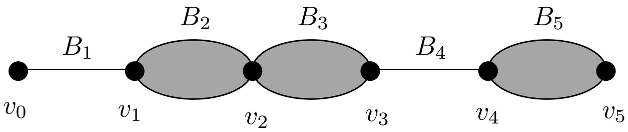


FIG. 1. *Example of a chain.*

no pair of edges crossing such that a, b, c, d occur on the boundary of the disc in this cyclic order.

For a graph G and $x, y \in V(G)$ let $G - xy$ denote the graph with vertex set $V(G)$ and edge set $E(G) - \{xy\}$. (Note that xy need not be an edge of G.)

DEFINITION 1.4. *Let G be a graph and let $H := v_0B_1v_1 \dots v_{k-1}B_kv_k$ be a chain. If H is an induced subgraph of G, then we say that H is a* chain in G. *We say that H is a* planar chain in G *if, for each $1 \leq i \leq k$ with $|V(B_i)| \geq 3$ (or equivalently, B_i is 2-connected), there exist distinct vertices $x_i, y_i \in V(G) - V(H)$ such that $(G[V(B_i) \cup \{x_i, y_i\}] - x_iy_i, x_i, v_{i-1}, y_i, v_i)$ is planar, and $B_i - \{v_{i-1}, v_i\}$ is a component of $G - \{x_i, y_i, v_{i-1}, v_i\}$. We also say that H is a* planar v_0-v_k chain. *See Figure* 2 *for two drawings of an example with $k = 5$. The dashed edges there may or may not exist, but they are not part of H.*

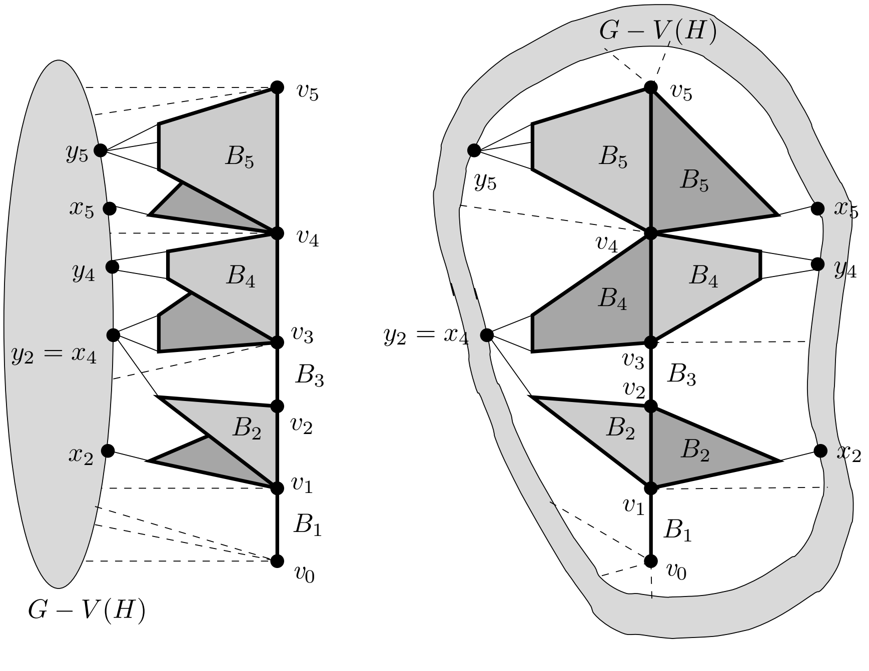


FIG. 2. *A planar chain $H := v_0B_1v_1B_2v_2B_3v_3B_4v_4B_5v_5$ in a graph G.*

DEFINITION 1.5. *Let G be a graph, let $S \subseteq V(G)$, and let k be a positive integer. We say that G is (k, S)-connected if $|V(G)| \geq |S| + 1$, G is connected, and, for any $T \subset V(G)$ with $|T| \leq k - 1$, every component of $G - T$ contains an element of S.*

This definition is partially motivated by the following observation. Let G be a k-connected graph, let $S \subseteq V(G)$, and let K be a component of $G - S$. Then $G[V(K) \cup S]$ is (k, S)-connected.

For a graph G and a subgraph H of G, we use $N_G(H)$ to denote the set of vertices in $V(G) - V(H)$ which are adjacent to at least one vertex in $V(H)$. Now we are ready to describe the main result of this paper. It is stated in a form which can be conveniently used in a forthcoming paper. (See Figure 5 for an illustration of the hypothesis of the theorem.)

THEOREM 1.6. *Let G be a graph, let a, b be distinct vertices of G, let P be a nonseparating induced path in G between a and b, let B_P be a nontrivial block of $G-V(P)$, and let $X_P:=N_G(G-V(B_P))$. Suppose $G-(V(B_P)-X_P)$ is $(4, X_P\cup\{a, b\})$-connected. Then there exists a planar a-b chain H in G such that $B_P \subseteq G-V(H)$ and $G-V(H)$ is 2-connected. Moreover, such a chain can be found in $O(|V(G)||E(G)|)$ time.*

There are two interesting consequences of Theorem 1.6 related to an open problem posed by Lovász [13]. (See section 4.)

COROLLARY 1.7. *Let G be a 4-connected graph, let a, b be distinct vertices of G, and let P be a nonseparating induced path in G between a and b such that $G-V(P)$ has a nontrivial block. Then there is a path Q between a and b in G such that $G-V(Q)$ is 2-connected, and such a path can be found in $O(|V(G)||E(G)|)$ time.*

COROLLARY 1.8. *Let G be a 4-connected graph and $ra \in E(G)$. Then there exists a cycle C in G through ra such that $G-(V(C)-\{r\})$ is 2-connected. Moreover, such a cycle can be found in $O(|V(G)|^2)$ time.*

The rest of this paper is organized as follows. In the remainder of this section we establish some notation we will use throughout the paper. In section 2 we give several auxiliary lemmas. These lemmas concern the existence of certain nonseparating paths in graphs with some connectivity constraints. In section 3 we prove Theorem 1.6. In section 4 we prove several consequences of Theorem 1.6, including Corollaries 1.7 and 1.8.

Throughout this paper, we use $A := B$ to rename B as A, or to define A as B.

Let G be a graph. For $S \subseteq V(G)$, let $N_G(S) := \{x \in V(G) - S : xy \in E(G),$ for some $y \in S\}$. Thus, for a subgraph H of G, $N_G(H) = N_G(V(H))$. When $S = \{x\}$, we let $N_G(x) := N_G(\{x\})$. When there exists no ambiguity, we may simply use $N(S), N(H)$, and $N(x)$ instead of $N_G(S), N_G(H)$, and $N_G(x)$, respectively. For a set F of 2-element subsets of $V(G)$, let $G+F$ denote the graph with vertex set $V(G)$ and edge set $E(G) \cup F$. If $F := \{xy\}$, let $G + xy := G + F$.

We describe a *path* in G as a sequence $P = (v_1, v_2, \ldots, v_k)$ of distinct vertices of G such that $v_iv_{i+1} \in E(G)$, $1 \le i \le k-1$. The vertices v_1 and v_k are called the *ends* of the path P, and the vertices in $V(P) - \{v_1, v_k\}$ are called the *internal vertices* of P. For $1 \le i \le j \le k$, let $P[v_i, v_j] := (v_i, \ldots, v_j)$, and for $1 \le i < j \le k$, let $P(v_i, v_j) := P[v_{i+1}, v_{j-1}]$. For $A, B \subseteq V(G)$, we say that a path P is an *A-B* path if one end of P is in A, the other end is in B, and no internal vertex of P is in $A \cup B$. If P is a path with ends a and b, we say that P is a *path from a to b*, or P is an *a-b* path. Two paths P and Q are *disjoint* if $V(P) \cap V(Q) = \emptyset$. Two paths are *internally disjoint* if no internal vertex of one is contained in the other. Given a path P in G and a set $S \subset V(G)$ (respectively, a subgraph S of G), we say that P is *internally disjoint from S* if no internal vertex of P is contained in S (respectively, $V(S)$). We also describe a *cycle* in G as a sequence $C = (v_1, v_2, \ldots, v_k, v_1)$ such that the vertices $v_1, \ldots, v_k$ are distinct, $v_iv_{i+1} \in E(G)$, for $1 \le i \le k-1$, and $v_kv_1 \in E(G)$.

2. Nonseparating paths.

In trying to find a nonseparating planar chain, we need to be able to find efficiently disjoint paths and nonseparating paths in graphs which satisfy certain connectivity conditions. The purpose of this section is to provide auxiliary lemmas (and algorithms) to deal with these problems.

The *disjoint paths problem* can be defined as follows. Given a graph G and distinct vertices a, b, c, d of G, find disjoint paths from a to b, and from c to d, respectively, or certify that they do not exist.

This problem was solved independently in [2, 16, 17, 19]. We state Seymour's version [16, Theorem 4.1].

THEOREM 2.1. *Let a, b, c, d be distinct vertices of a graph G. Then exactly one of the following holds:*

(1) *G contains disjoint paths from a to b and from c to d, respectively, or*

(2) *for some integer $k \geq 0$, there exist pairwise disjoint sets $A_1, \ldots, A_k \subseteq V(G) - \{a, b, c, d\}$ such that*
   - *for $1 \leq i \neq j \leq k$, $N_G(A_i) \cap A_j = \emptyset$,*
   - *for $1 \leq i \leq k$, $|N_G(A_i)| \leq 3$, and*
   - *if G' is the graph obtained from G by, for each i, deleting A_i and adding new edges joining every pair of distinct vertices in $N_G(A_i)$, and also adding the edges ab and cd, then G' can be drawn in the plane with no pair of edges crossing except ab and cd, which cross once.*

Let G be a graph and $S := \{a, b, c, d\} \subseteq V(G)$. Shiloach [17] gave an $O(|V(G)||E(G)|)$ algorithm for the disjoint paths problem. We need to solve a special case of the disjoint paths problem, namely, when G is $(4, S)$-connected. We show in the appendix that Shiloach's algorithm can solve the disjoint paths problem in $O(|V(G)| + |E(G)|)$ time for $(4, S)$-connected graphs.

LEMMA 2.2. *Let G be a graph and let $S := \{a, b, c, d\} \subset V(G)$. Suppose that G is $(4, S)$-connected. Then exactly one of the following holds:*

(1) *there exist disjoint paths from a to b and from c to d, respectively, or*

(2) *(G, a, c, b, d) is planar.*

Moreover, one can in $O(|V(G)| + |E(G)|)$ time find paths as in (1) *or certify that* (2) *holds.*

The rest of this section deals with nonseparating induced paths in graphs with certain connectivity properties. We also show how to find these paths efficiently.

LEMMA 2.3. *Let G be a connected graph, $S \subseteq V(G)$, $\{a, a'\} \subseteq S$, and let P be an a-a' path in G. Suppose*

(i) *G is $(3, S)$-connected, and*

(ii) *$S - \{a, a'\}$ is contained in a component U of $G - V(P)$.*

Then there exists a nonseparating induced a-a' path P' in G such that $V(P') \cap V(U) = \emptyset$. Moreover, such a path can be found in $O(|V(G)| + |E(G)|)$ time.

Proof. We may assume that P is induced; otherwise, we can find in $O(|V(G)| + |E(G)|)$ time an induced a-a' path in G satisfying (ii). If P is nonseparating, then $P' := P$ is the required path. If $|V(P)| = 2$, then by (i) every component of $G - V(P)$ contains a vertex of S, and so by (ii) $G - V(P) = U$, which implies that P is nonseparating. So we may assume that $|V(P)| \geq 3$ and $G - V(P)$ is not connected.

Let G' be the graph obtained from G by contracting U to a single vertex u, adding the edges aa', ua, and ua' and removing multiple edges. See Figure 3. Note that a, a' belong to the cycle $P + aa'$. We claim that $H := G' - u$ is 2-connected. Suppose for a contradiction that there exists $v \in V(H)$ such that $H - v$ is disconnected. Since a, a' are vertices of $P + aa'$ which is a cycle in H, there exists a component K in $H - v$ which does not contain any vertex of P. But then K is a component of $G - v$ which does not contain any vertex in S, contradicting (i). Thus, $G' - u$ is 2-connected.

In fact, G' must be 3-connected. Suppose for a contradiction that G' is not 3-connected. Then there is a vertex cut T in G' with $|T| \leq 2$. Since $G' - u$ is

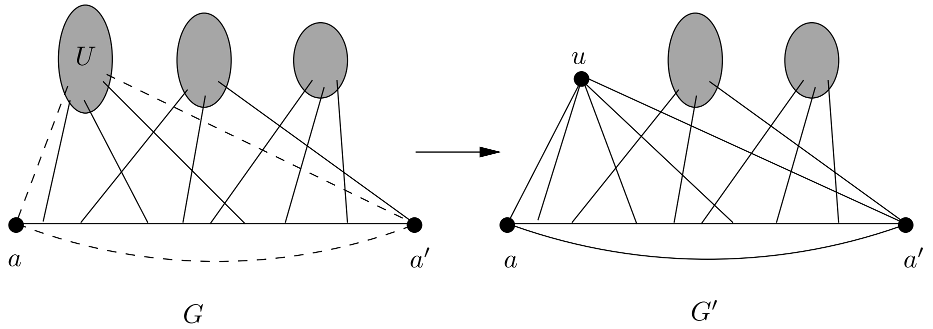


FIG. 3. *Graphs G and G' in the proof of Lemma* 2.3.

2-connected, $u \notin T$. Moreover, since $\{u, a, a'\}$ induces a triangle in G', there exists a component K of $G' - T$ which does not contain any of u, a, a'. But then K is also a component of $G - T$ which does not contain any element of S, contradicting (i). Hence, G' is 3-connected.

By Theorem 1.2 (with G', aa', u as G, e, u, respectively), there exists a nonseparating induced cycle C in G' containing aa' and avoiding u. Moreover, such a cycle can be found in $O(|V(G')| + |E(G')|)$ time (and hence in $O(|V(G)| + |E(G)|)$ time). Thus, $P' := C - aa'$ is a nonseparating induced path in G such that $V(P') \cap V(U) = \emptyset$. □

As an application of Lemma 2.3, we derive the following strengthening of Lemma 2.2.

LEMMA 2.4. *Let G be a graph and $S := \{a, a', b, b'\} \subseteq V(G)$. Suppose that G is $(4, S)$-connected. Then exactly one of the following holds:*

(1) *there exists a nonseparating induced a-a' path P' in G such that $V(P') \cap \{b, b'\} = \emptyset$, or*

(2) *(G, a, b, a', b') is planar.*

Moreover, one can in $O(|V(G)| + |E(G)|)$ time find a path as in (1) *or certify that* (2) *holds.*

Proof. By Lemma 2.2, either (a) there exist disjoint paths P and Q in G from a to a' and from b to b', respectively, or (b) (G, a, b, a', b') is planar. Moreover, one can in $O(|V(G)| + |E(G)|)$ time find paths as in (a) or certify that (b) holds. If (b) holds, then (2) holds. Assume (a) holds. Let U be the component of $G - V(P)$ containing $S - \{a, a'\} = \{b, b'\}$. Since G is $(4, S)$-connected (and hence $(3, S)$-connected), G, P, S, U and $\{a, a'\}$ satisfy the hypothesis of Lemma 2.3. Thus, by Lemma 2.3 there exists a nonseparating a-a' path P' in G such that $V(P') \cap V(U) = \emptyset$, and such a path can be found in $O(|V(G)| + |E(G)|)$ time. Hence, $V(P') \cap \{b, b'\} = \emptyset$, and P' satisfies (1). □

To prove the final result of this section, we need the following result of Cheriyan and Maheshwari [3, p. 516], which is the core of the linear algorithm in [3] for finding a nonseparating induced cycle as described in Theorem 1.2.

THEOREM 2.5. *Let G be a 3-connected graph, let $aa' \in E(G)$, and let C be a nonseparating induced cycle in G containing aa'. Then there exists another nonseparating induced cycle C' in G such that $V(C') \cap V(C) = \{a, a'\}$ and $E(C') \cap E(C) = \{aa'\}$. Moreover, such a cycle can be found in $O(|V(G)| + |E(G)|)$ time.*

Our next result is in the same spirit as Theorem 2.5, but we relax the 3-connectivity condition. Therefore, it is more convenient to use.

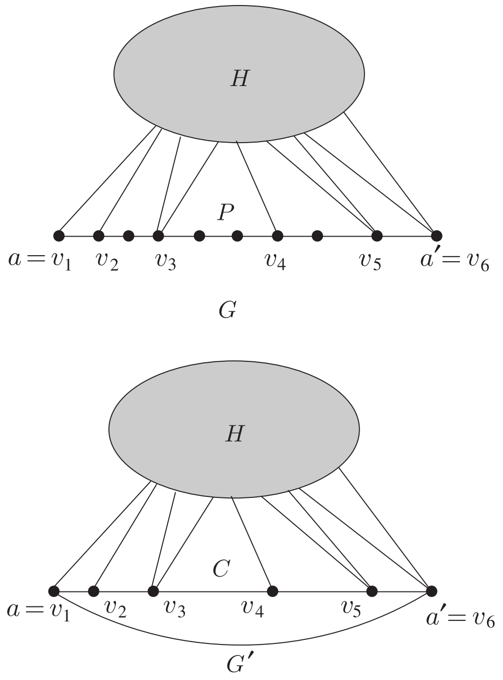


FIG. 4. *G and G' as in the proof of Lemma* 2.6.

LEMMA 2.6. *Let G be a connected graph, let a, a' be distinct vertices of G with degree at least two, and let P be a nonseparating induced a-a' path in G. Suppose that G is $(3, V(P))$-connected. Then there exists another nonseparating induced a-a' path P' in G such that $V(P') \cap V(P) = \{a, a'\}$ and $E(P') \cap E(P) = \emptyset$. Moreover, such a path can be found in $O(|V(G)| + |E(G)|)$ time.*

Proof. For convenience, let $H := G - V(P)$. Since P is a nonseparating path in G, H is connected. Moreover, both a and a' have a neighbor in $V(H)$ because both have degree at least two in G and P is induced. Let $v_1 = a, v_2, \ldots, v_k = a'$ be the neighbors of H on P in this order from a to a'. (See Figure 4 for an illustration.) Note that $k \geq 3$ because G is $(3, V(P))$-connected. Let G' be the graph obtained from G by adding the edge aa' and by replacing, for each $1 \leq i \leq k-1$, the path $P[v_i, v_{i+1}]$ by an edge $v_i v_{i+1}$. Note that $C := G' - V(H)$ is a cycle in G'. See again Figure 4.

We claim that G' is 3-connected. Suppose for a contradiction that G' is not 3-connected. Then there is a vertex cut T in G' with $|T| \leq 2$. Note that $T \not\subseteq V(C)$, since H is connected and every vertex of C has a neighbor in H. But then $G' - T$ has a component K such that $V(K) \cap V(C) = V(K) \cap V(P) = \emptyset$. Hence, K is also a component of $G - T$ with $V(K) \cap V(P) = \emptyset$, contradicting the assumption that G is $(3, V(P))$-connected. Hence, G' is 3-connected.

By Theorem 2.5 (with G', C, a, a' as G, C, a, a', respectively), there exists a nonseparating cycle C' in G' such that $V(C') \cap V(C) = \{a, a'\}$ and $E(C') \cap E(C) = \{aa'\}$. Moreover, such a cycle can be found in $O(|V(G')| + |E(G')|)$ time (and hence in $O(|V(G)| + |E(G)|)$ time). Thus, $P' := C' - aa'$ is a nonseparating induced a-a' path in G such that $V(P') \cap V(P) = \{a, a'\}$ and $E(P') \cap E(P) = \emptyset$. □

3. Nonseparating chains. The main goal for this section is to design an algorithm that solves the following problem. Given G, a, b, P, B_P as in Theorem 1.6,

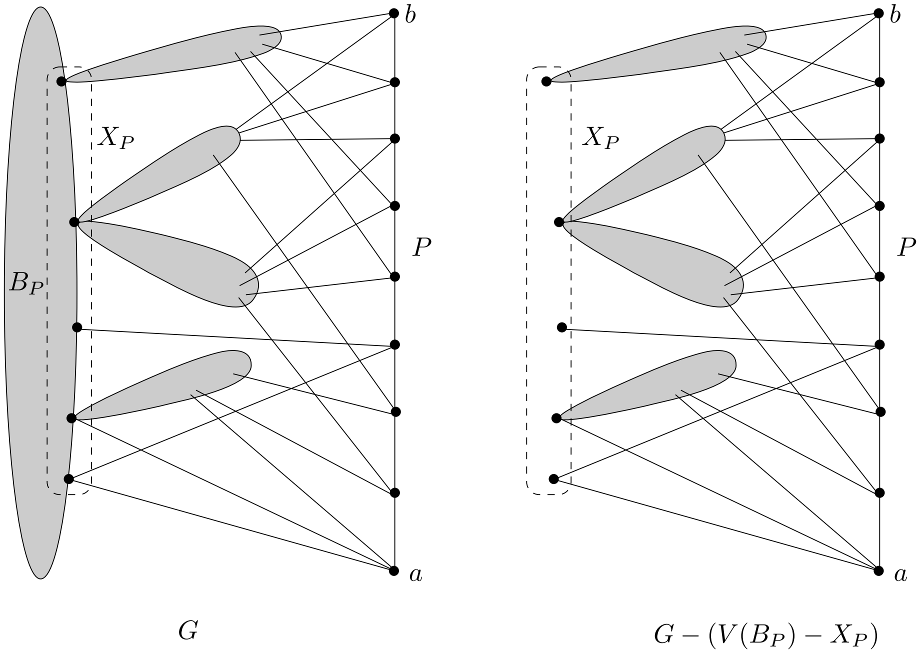


FIG. 5. *G, a, b, P, B_P, X_P in Notation* 3.1.

find a planar a-b chain H in G such that $G - V(H)$ is 2-connected and $V(B_P) \subseteq V(G) - V(H)$. For convenience, we fix the following notation throughout this section.

NOTATION AND ASSUMPTION 3.1. *Let G be a graph, let a, b be distinct vertices of G, let P be a nonseparating induced a-b path in G, let B_P be a nontrivial block of $G - V(P)$, and let $X_P := N_G(G - V(B_P))$. Suppose that $G - (V(B_P) - X_P)$ is $(4, X_P \cup \{a, b\})$-connected. See Figure* 5.

*Let $\mathcal{P}_P$ be the set of nonseparating induced a-b paths P' in G with $B_P \subseteq G - V(P')$. Note that $P \in \mathcal{P}_P$. For each $P' \in \mathcal{P}_P$ let $B_{P'}$ denote the nontrivial block of $G - V(P')$ containing B_P. We say that $P' \in \mathcal{P}_P$ is a B_P-*augmenting path *if $|V(B_P)| < |V(B_{P'})|$.*

Note that X_P consists of cut vertices of $G - V(P)$ contained in $V(B_P)$ and the neighbors of $V(P)$ contained in $V(B_P)$. Also note that our next result shows that the paths in $\mathcal{P}_P$ are well behaved.

LEMMA 3.2. *Let $P' \in \mathcal{P}_P$. Let $X_{P'} := N(G-V(B_{P'}))$. Then $G-(V(B_{P'})-X_{P'})$ is $(4, X_{P'} \cup \{a, b\})$-connected.*

Proof. For convenience, let $G' := G-(V(B_{P'})-X_{P'})$. Suppose for a contradiction that G' is not $(4, X_{P'} \cup \{a, b\})$-connected. Then there exists some $T \subset V(G')$ with $|T| \leq 3$ and there exists some component K of $G'-T$ such that $V(K) \cap (X_{P'} \cup \{a, b\}) = \emptyset$. Since $V(B_P) \subseteq V(B_{P'})$, for each $x \in X_P$, either $x \notin V(G')$ or $x \in X_{P'}$. Thus, $V(K) \cap X_P = \emptyset$. But then, K is a component of $(G - (V(B_P) - X_P)) - T$ which does not contain any vertex in $X_P \cup \{a, b\}$. This contradicts the assumption that $G - (V(B_P) - X_P)$ is $(4, X_P \cup \{a, b\})$-connected. Therefore, G' is $(4, X_{P'} \cup \{a, b\})$-connected. ▯

Let us describe the basic idea of the algorithm we want to design. At the beginning of each iteration we have a nonseparating a-b path P and a nontrivial block B_P of $G - V(P)$. The algorithm then tries to find a B_P-augmenting path $P' \in \mathcal{P}_P$. If the

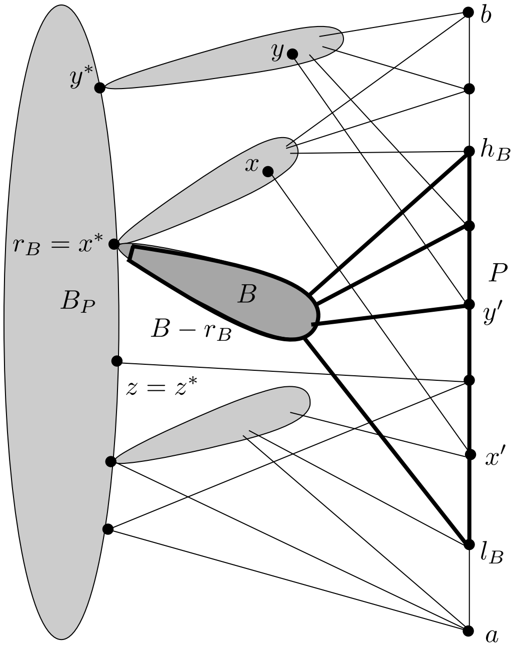


FIG. 6. *A nice bridge B and the graph G_B as defined in the proof of Lemma* 3.7 *shown in boldface.*

algorithm finds such a path P', then it starts a new iteration with P' as P. (Note that by Lemma 3.2, $G - (V(B_{P'}) - X_{P'})$ is $(4, X_{P'} \cup \{a, b\})$-connected.) If the algorithm does not find a B_P-augmenting path, then it finds a planar a-b chain as required in Theorem 1.6.

To describe this algorithm more precisely, we need more concepts and notation.

DEFINITION 3.3. *Let F be a subgraph of a graph K. An F-*bridge *of K is a subgraph of K which is induced by either* (1) *an edge in $E(K) - E(F)$ with both ends on F or* (2) *edges of a component D of $K - V(F)$ together with the edges of K from D to F. For an F-bridge B of K, the set $V(B) \cap V(F)$ is the set of* attachments *of B on F.*

NOTATION 3.4. *Let $\mathcal{B}$ denote the set of B_P-bridges of $G - V(P)$. For each $B \in \mathcal{B}$, $V(B_P) \cap V(B)$ consists of exactly one vertex (which is contained in X_P), and we let r_B denote this vertex. For any $x, y \in V(P)$, we denote $x \leq y$ if $x \in V(P[a, y])$. If $x \leq y$ and $x \neq y$, then we write $x < y$. In this case, we say that x is lower than y, or y is higher than x. Since G is $(4, X_P \cup \{a, b\})$-connected, for each $B \in \mathcal{B}$, $B - r_B$ has at least three neighbors on P. Let l_B and h_B denote the lowest and the highest neighbor of $B - r_B$ on P, respectively. See Figure* 6 *for an example.*

LEMMA 3.5. *The following hold:*

(1) *$V(P(l_B, h_B)) \neq \emptyset$ and $N_G(P(l_B, h_B)) \cap (V(B) - \{r_B\}) \neq \emptyset$, and*

(2) *$N_G(P(l_B, h_B)) \not\subseteq V(B) \cup V(P)$.*

Proof. (1) holds because $B - r_B$ has at least three neighbors on P, and (2) holds because P is an induced path in G and $\{r_B, l_B, h_B\}$ is not a 3-vertex cut of G. □

Next, we describe members of $\mathcal{B}$ which we can use to produce a B_P-augmenting path.

DEFINITION 3.6. *For each vertex x of $G - V(P)$, we define x^* as follows. If $x \in V(B)$ for some $B \in \mathcal{B}$, then let $x^* := r_B$. If $x \in V(B_P)$, then $x^* := x$. We say* *that a member B of $\mathcal{B}$ is a* nice bridge *if there exist $x, y \in N_G(P(l_B, h_B)) - ((V(B) - \{r_B\}) \cup V(P))$ such that $x^* \neq y^*$. See Figure* 6 *for an example.*

The next lemma shows that any nice bridge can be used to find a B_P-augmenting path.

Lemma 3.7. *Let $B \in \mathcal{B}$ be a nice bridge. Then there exists an induced l_B-h_B path Q in $G[V(B) \cup \{l_B, h_B\}]$ such that $P' := (P - V(P(l_B, h_B))) \cup Q$ is a B_P-augmenting path in G. Moreover, such a path Q can be found in $O(|V(G)| + |E(G)|)$ time.*

Proof. Since B is a nice bridge, there exist $x, y \in N_G(P(l_B, h_B)) - ((V(B) - \{r_B\}) \cup V(P))$ such that $x^* \neq y^*$. See Figure 6. Let G_B be the subgraph of G induced by $(V(B) - \{r_B\}) \cup V(P[l_B, h_B])$. Since B is a B_P-bridge, $B - r_B$ is connected. Thus, $P[l_B, h_B]$ is a nonseparating induced path in G_B. Furthermore, since G is $(4, X_P \cup \{a, b\})$-connected, for any $T \subset V(G_B)$ with $|T| \leq 2$, every component of $G_B - T$ contains a vertex of $V(P[l_B, h_B])$. (Otherwise, $T \cup \{r_B\}$ is a 3-cut of G, and $G - (T \cup \{r_B\})$ has a component not containing any element of $X_P \cup \{a, b\}$.) Thus, G_B is $(3, V(P[l_B, h_B]))$-connected. By Lemma 2.6 (with $G_B, l_B, h_B, P[l_B, h_B]$ as G, a, a', P, respectively), there exists a nonseparating induced l_B-h_B path Q in G_B disjoint from $P(l_B, h_B)$. Moreover, such a path Q can be found in $O(|V(G_B)| + |E(G_B)|)$ time. Since $|V(G_B)| + |E(G_B)| = O(|V(G)| + |E(G)|)$, such a path Q can be found in $O(|V(G)| + |E(G)|)$ time.

Clearly, the path $P' = (P - V(P(l_B, h_B))) \cup Q$ is an induced a-b path in G.

Let us prove that P' is nonseparating in G. It suffices to prove that for every $v \notin V(B_P) \cup V(P')$, there exists a $\{v\}$-$V(B_P)$ path in $G - V(P')$. First, suppose $v \in V(B')$ for some $B' \in \mathcal{B}$ with $B' \neq B$. Since $V(B') \cap V(P') = \emptyset$, there exists a v-$r_{B'}$ path in B' (and hence in $G - V(P')$). So we may assume $v \in (V(B) - \{r_B\}) \cup V(P(l_B, h_B))$. Since $N_G(P(l_B, h_B)) \not\subseteq V(B) \cup V(P)$ (by (2) of Lemma 3.5) and $V(Q(l_B, h_B)) \cap V(P(l_B, h_B)) = \emptyset$, and because Q is a nonseparating path in G_B, there exists a v-$V(B_P)$ path in $G - V(P')$. Hence, P' is nonseparating in G.

Thus, $P' \in \mathcal{P}_P$. It remains for us to show that $|V(B_P)| < |V(B_{P'})|$. Note that G contains disjoint paths P_x and P_y from x to x^* and from y to y^*, respectively, and P_x and P_y are disjoint from $P \cup (B - r_B) \cup (B_P - \{x^*, y^*\})$. Let $x', y' \in V(P(l_B, h_B))$ such that $xx', yy' \in E(G)$. Then both B_P and the path $(P_x \cup P[x', y'] \cup P_y) + \{xx', yy'\}$ are contained in $B_{P'}$. Hence, $|V(B_P)| < |V(B_{P'})|$, and so, P' is a B_P-augmenting path. ∎

In what follows we prove several lemmas which will help us find nice bridges (and hence, B_P-augmenting paths by Lemma 3.7). But first, we need the following.

Definition 3.8. *We say that two B_P-bridges B and B' in $\mathcal{B}$ are* overlapping *if the paths $P[l_B, h_B]$ and $P[l_{B'}, h_{B'}]$ have an edge in common. Define an auxiliary graph $\mathcal{K}$ such that $V(\mathcal{K}) = \mathcal{B}$, and B and B' are adjacent in $\mathcal{K}$ if B and B' are overlapping. See Figure* 7 *for an example.*

The next two lemmas appear in [5]. Since their proofs are short, we include them here.

Lemma 3.9. *Let (B_1, B_2, B_3) be an induced path in $\mathcal{K}$ such that $r_{B_1} \neq r_{B_3}$. Then B_2 is a nice bridge.*

Proof. By the definition of $\mathcal{K}$ and by the assumption that (B_1, B_2, B_3) is induced in $\mathcal{K}$, B_1 and B_3 are not overlapping. Thus we may assume $l_{B_1} < h_{B_1} \leq l_{B_3} < h_{B_3}$. Moreover, $l_{B_2} < h_{B_1}$ and $l_{B_3} < h_{B_2}$. Let $x \in V(B_1) - \{r_{B_1}\}$ such that $xh_{B_1} \in E(G)$ and let $y \in V(B_3) - \{r_{B_3}\}$ such that $yl_{B_3} \in E(G)$. Clearly, $x, y \in N_G(P(l_{B_2}, h_{B_2}))$, $x, y \notin (V(B_2) - \{r_{B_2}\}) \cup V(P)$, and $x^* = r_{B_1} \neq r_{B_3} = y^*$. Hence by Definition 3.6, B_2 is a nice bridge. ∎

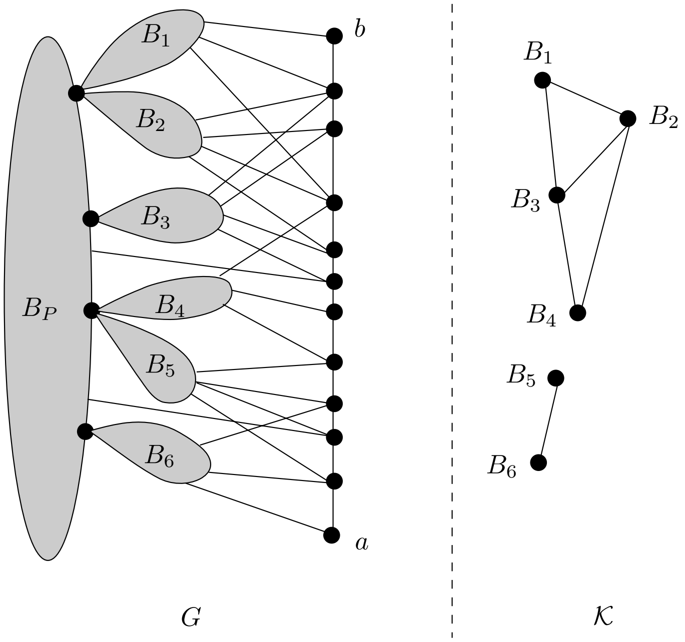


FIG. 7. *An example of an auxiliary graph $\mathcal{K}$.*

LEMMA 3.10. *Let (B_1, B_2, B_3) be a path in $\mathcal{K}$ such that $r_{B_1} \neq r_{B_2} \neq r_{B_3} \neq r_{B_1}$. Then one can find in constant time some $i \in \{1, 2, 3\}$ such that B_i is a nice bridge.*

Proof. If the path (B_1, B_2, B_3) is induced in $\mathcal{K}$, then the result follows from Lemma 3.9. So suppose that B_1, B_2, B_3 induces a triangle in $\mathcal{K}$. By symmetry, assume that $P[l_{B_1}, h_{B_1}]$ is not properly contained in $P[l_{B_i}, h_{B_i}]$ for $i = 2, 3$ (this can be checked in constant time). Thus, for each $i \in \{2, 3\}$, either $l_{B_i} \in V(P(l_{B_1}, h_{B_1}))$ or $h_{B_i} \in V(P(l_{B_1}, h_{B_1}))$ or $P[l_{B_1}, h_{B_1}] = P[l_{B_i}, h_{B_i}]$. Therefore, since $N_G(P(l_{B_i}, h_{B_i})) \cap (V(B_i) - \{r_{B_i}\}) \neq \emptyset$ (by (1) of Lemma 3.5), it follows that there exist $x \in N_G(P(l_{B_1}, h_{B_1})) \cap (V(B_2) - \{r_{B_2}\})$ and $y \in N_G(P(l_{B_1}, h_{B_1})) \cap (V(B_3) - \{r_{B_3}\})$. Note that $x, y \notin (V(B_1) - \{r_{B_1}\}) \cup V(P)$, and $x^* = r_{B_2} \neq r_{B_3} = y^*$. Hence by Definition 3.6, B_1 is a nice bridge. ☐

To find B_P-augmenting paths, we need to search the components of $\mathcal{K}$. For convenience, we introduce the following notation.

NOTATION 3.11. *Let $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_t$ be the components of the auxiliary graph $\mathcal{K}$. For $j = 1, \ldots, t$ let $V_j := \bigcup_{B \in V(\mathcal{A}_j)} V(B)$, let $Q_j := \bigcup_{B \in V(\mathcal{A}_j)} P[l_B, h_B]$, and let $R_{\mathcal{A}_j} := \{r_B : B \in V(\mathcal{A}_j)\}$. Note that V_j is a subset of $V(G) - (V(B_P) - X_P)$, Q_j is a subpath of P, and $R_{\mathcal{A}_j} \subseteq X_P$.*

The number of edges in a component of $\mathcal{K}$ can be $O(|V(\mathcal{K})|^2)$, but for our purpose, we need to compute only a spanning tree of each component.

LEMMA 3.12. *Algorithm* 1 *constructs rooted spanning trees $\mathcal{T}_j$ of $\mathcal{A}_j$ for all $j = 1, \ldots, t$, and finds the ends a_j, b_j of Q_j with $a_j < b_j$, for all $j = 1, \ldots, t$. Furthermore, for any $j \in \{1, \ldots, t\}$ and any element B of $V(\mathcal{T}_j)$, the path from the root of $\mathcal{T}_j$ to B in $\mathcal{T}_j$ is induced in $\mathcal{K}$. Moreover, all $\mathcal{T}_j, a_j, b_j$ can be found in $O(|V(G)| + |E(G)|)$ time.*

Proof. The set $\mathcal{Q}$ is implemented as a queue, and for each vertex x of the path P we keep a list of B_P-bridges B of $G - V(P)$ such that $l_B = x$. The index k is used to avoid rescanning a vertex more than once. The algorithm is basically a variation of

Algorithm 1. Construct forest.

```
Require: The set $\mathcal{B}$ of $B_P$-bridges of $G - V(P)$.
Return: An integer $t \geq 0$, spanning trees $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_t$ of the components of the
  auxiliary graph $\mathcal{K}$, and the ends $a_j, b_j$ of $Q_j$ with $a_j < b_j$ for $j = 1, \ldots, t$.
  Let $P = (a = x_1, x_2, \ldots)$;
  $j \leftarrow 1$;
  $k \leftarrow 1$;
  while $k \leq |V(P)|$ do
    Let $B \in \mathcal{B}$ such that $l_B = x_k$;
    $\mathcal{T}_j \leftarrow B$;
    $a_j \leftarrow l_B$;
    $b_j \leftarrow h_B$;
    $\mathcal{Q} \leftarrow \{B\}$;
    while $\mathcal{Q} \neq \emptyset$ do
      Let $B' \in \mathcal{Q}$;
      $\mathcal{Q} \leftarrow \mathcal{Q} - \{B'\}$;
      Let $x_{k'} = h_{B'}$;
      for $i \leftarrow k$ to $k' - 1$ do
        for each $B$ such that $l_B = x_i$ do
          if $B \notin V(\mathcal{T}_j)$ then
            $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{B\}$;
            $\mathcal{T}_j \leftarrow (\mathcal{T}_j \cup \{B\}) + BB'$;
      if $k' > k$ then
        $k \leftarrow k'$;
    $b_j \leftarrow x_k$;
    while $k \leq |V(P)|$ and there exists no $B \in \mathcal{B}$ such that $l_B = x_k$ do
      $k \leftarrow k + 1$;
    $j \leftarrow j + 1$;
```

the breadth-first search method and can be implemented to run in $O(|V(G)|)$ time. It is easy to see that each $\mathcal{T}_j$ is a spanning tree of a component of $\mathcal{K}$. The first vertex inserted in $\mathcal{T}_j$ becomes its root. Furthermore, it is not hard to see that (from the nature of breadth-first search) $\mathcal{T}_j$ satisfies the following property: for any element B of $V(\mathcal{T}_j)$, the path in $\mathcal{T}_j$ from B to the root of $\mathcal{T}_j$ is induced in $\mathcal{K}$. □

From Notation 3.11 and Algorithm 1, we see that $Q_j = P[a_j, b_j]$.

Next, for a component $\mathcal{A}_j$ of $\mathcal{K}$ (or more precisely, the spanning tree $\mathcal{T}_j$ computed by Algorithm 1) we derive necessary and sufficient conditions for the existence of a nice bridge in $V(\mathcal{A}_j)$, and hence we may apply Lemma 3.7 to derive the existence of a B_P-augmenting path. We do this by considering the size of $R_{\mathcal{A}_j}$.

Lemma 3.13. *Let $\mathcal{A}_j$ be a component of $\mathcal{K}$ such that $|R_{\mathcal{A}_j}| \geq 3$. Then there exists a member of $V(\mathcal{A}_j)$ which is a nice bridge. Moreover, such a member of $V(\mathcal{A}_j)$ can be found in $O(|V(G)|)$ time.*

Proof. We want to show that $\mathcal{T}_j$ contains a path (B_1, B_2, B_3) such that either (i) (B_1, B_2, B_3) is an induced path in $\mathcal{K}$ and $r_{B_1} \neq r_{B_3}$ or (ii) $r_{B_1} \neq r_{B_2} \neq r_{B_3} \neq r_{B_1}$. For convenience, let $\mathcal{T} := \mathcal{T}_j$. Since $|R_{\mathcal{A}_j}| \geq 3$, there exist members W, Y, and Z of $V(\mathcal{A}_j)$ such that $r_W \neq r_Y \neq r_Z \neq r_W$. Moreover, W, Y, and Z can be found in $O(|V(\mathcal{T})|)$ time and hence in $O(|V(G)|)$ time. We may assume that W is the root of $\mathcal{T}$. By Lemma 3.12, $\mathcal{T}[W, Y]$ and $\mathcal{T}[W, Z]$ are induced paths in $\mathcal{A}_j$.

Suppose neither $\mathcal{T}[W,Y]$ nor $T[W,Z]$ contains a path (B_1,B_2,B_3) satisfying (i) or (ii) above. Because $r_W \neq r_Y$, $r_B \in \{r_W, r_Y\}$ for every member B of $V(\mathcal{T}[W,Y])$ and $r_{B_1} \neq r_{B_2}$ for every member B_1B_2 of $E(\mathcal{T}[W,Y])$. Similarly, because $r_W \neq r_Z$, $r_B \in \{r_W, r_Z\}$ for every member of $V(\mathcal{T}[W,Z])$ and $r_{B_1} \neq r_{B_2}$ for any member B_1B_2 of $E(\mathcal{T}[W,Z])$. But since r_Z is distinct from r_W and r_Y, it follows that $\mathcal{T}[W,Y] \cup \mathcal{T}[W,Z]$ must contain a path (B_1,B_2,B_3) which satisfies (i) or (ii). Clearly, this path can be found in $O(|V(G)|)$ time.

By Lemmas 3.9 and 3.10, one of B_1,B_2,B_3 is a nice bridge, and such a bridge can be found in $O(|V(G)|)$ time. □

If $|R_{\mathcal{A}_j}| \leq 2$ for every $j \in \{1,\dots,t\}$, then the existence of a nice bridge is not guaranteed. In this case, we will find certain 4-cuts of G which play a fundamental role in the construction of the desired planar a-b chain.

LEMMA 3.14. *Let $\mathcal{A}_j$ be a component of $\mathcal{K}$ such that $|R_{\mathcal{A}_j}| = 1$. Then one of the following holds:*

(1) $|V(\mathcal{A}_j)| = 1$ *and* $|(X_P \cap N_G(Q_j(a_j,b_j))) - R_{\mathcal{A}_j}| = 1$, *or*

(2) *a member of* $V(\mathcal{A}_j)$ *is a nice bridge, and it can be found in* $O(|V(G)|)$ *time.*

Proof. We claim that $(X_P \cap N_G(Q_j(a_j,b_j))) - R_{\mathcal{A}_j} \neq \emptyset$. Otherwise, there exists a component of $G - (R_{\mathcal{A}_j} \cup \{a_j,b_j\})$ not containing any element of $X_P \cup \{a,b\}$ (because P is induced), which is a contradiction to the assumption that G is $(4, X_P \cup \{a,b\})$-connected. Thus, let $x \in (X_P \cap N_G(Q_j(a_j,b_j))) - R_{\mathcal{A}_j}$.

First, suppose $|V(\mathcal{A}_j)| \geq 2$. Let $B \in V(\mathcal{A}_j)$ such that $x \in N_G(P(l_B,h_B))$. Since $|V(\mathcal{A}_j)| \geq 2$, there exists $B' \in V(\mathcal{A}_j)$ such that $B' \neq B$, and B and B' overlap. By renaming B and B' if necessary, we may assume that $P[l_B,h_B]$ is not a proper subpath of $P[l_{B'},h_{B'}]$. Then, either $l_{B'} \in V(P(l_B,h_B))$, or $h_{B'} \in V(P(l_B,h_B))$, or both $l_B = l_{B'}$ and $h_B = h_{B'}$. By (1) of Lemma 3.5, $N_G(P(l_{B'},h_{B'})) \cap (V(B') - \{r_{B'}\}) \neq \emptyset$. Hence, $P(l_B,h_B)$ has a neighbor y such that $y \in V(B') - \{r_{B'}\}$. Note that $x, y \in N_G(P(l_B,h_B))$, $x, y \notin (V(B) - \{r_B\}) \cup V(P)$, $x^* = x \notin V_j$ and $y^* = r_{B'} \in V_j$. Thus by Definition 3.6, B is a nice bridge. Clearly, B can be found in $O(|V(G)|)$ time, and hence, (2) holds.

Now, assume that $|V(\mathcal{A}_j)| = 1$ and B is the only member of $V(\mathcal{A}_j)$. Then $Q_j = P[l_B,h_B]$. Suppose (1) does not hold. Then $|(X_P \cap N_G(P(l_B,h_B))) - R_{\mathcal{A}_j}| > 1$. Hence, there exists some $y \in (X_P \cap N_G(P(l_B,h_B))) - R_{\mathcal{A}_j}$ with $y \neq x$. Then $x, y \in N_G(P(l_B,h_B))$, $x, y \notin (V(B) - \{r_B\}) \cup V(P)$, and $x^* = x \neq y = y^*$. Hence by Definition 3.6, B is a nice bridge. Again, (2) holds. □

LEMMA 3.15. *Let $\mathcal{A}_j$ be a component of $\mathcal{K}$ such that $|R_{\mathcal{A}_j}| = 2$. Then one of the following holds:*

(1) $X_P \cap N_G(Q_j(a_j,b_j)) \subseteq R_{\mathcal{A}_j}$, *or*

(2) *a member of* $V(\mathcal{A}_j)$ *is a nice bridge, and it can be found in* $O(|V(G)|)$ *time.*

Proof. Suppose that (1) does not hold. Then there exists some $x \in (X_P \cap N_G(Q_j(a_j,b_j))) - R_{\mathcal{A}_j}$. Note that $x^* = x$. Let $B \in V(\mathcal{A}_j)$ such that $x \in N_G(P(l_B,h_B))$. Since $|R_{\mathcal{A}_j}| = 2$, we have $|V(\mathcal{A}_j)| \geq 2$, and hence there exists $B' \in V(\mathcal{A}_j)$ such that $B' \neq B$, and B and B' overlap. We can rename B and B' if necessary so that $P[l_B,h_B]$ is not a proper subpath of $P[l_{B'},h_{B'}]$. We can show that B is a nice bridge as in the second paragraph in the proof of Lemma 3.14. □

Before we can fully describe the main algorithm, we need to deal with the situation where (1) of Lemma 3.14 or (1) of Lemma 3.15 occurs.

DEFINITION 3.16. *Let $\mathcal{A}_j$ be a component of $\mathcal{K}$ such that either* (i) $|R_{\mathcal{A}_j}| = 1$ *and* $|(X_P \cap N_G(Q_j(a_j,b_j))) - R_{\mathcal{A}_j}| = 1$, *or* (ii) $|R_{\mathcal{A}_j}| = 2$ *and* $N_G(X_P \cap Q_j(a_j,b_j)) \subseteq R_{\mathcal{A}_j}$. *If* (i) *holds, then let* $R_{\mathcal{A}_j} := \{c_j\}$, *let* $(X_P \cap N(Q_j(a_j,b_j))) - R_{\mathcal{A}_j} := \{d_j\}$, *and let*

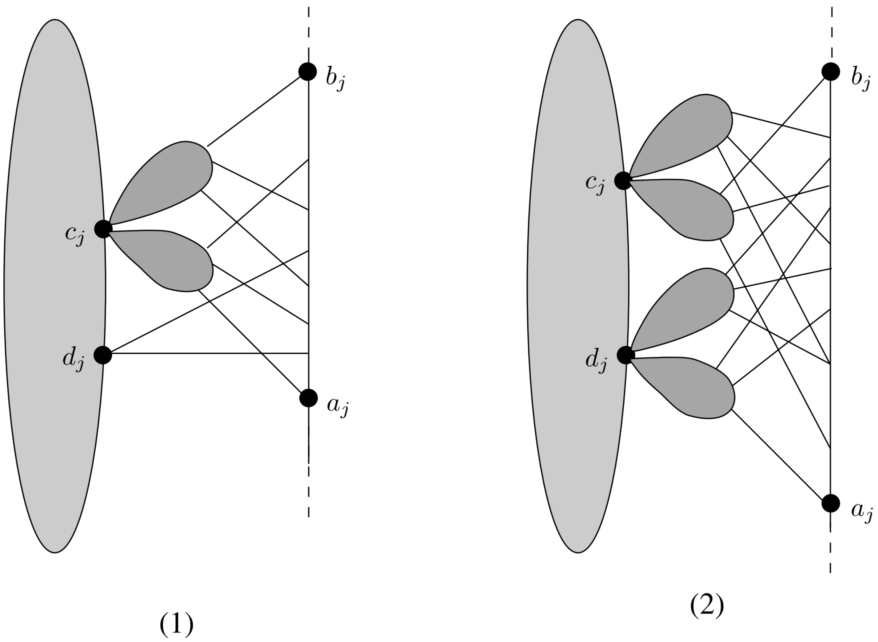


FIG. 8. *4-cuts determined by a component $\mathcal{A}_j$ of $\mathcal{K}$.*

$G_j := G[V_j \cup \{d_j\} \cup V(Q_j)] - c_jd_j$. If (ii) *holds, then let $R_{\mathcal{A}_j} := \{c_j, d_j\}$, and let $G_j := G[V_j \cup V(Q_j)] - c_jd_j$. In both cases, the set $S_j := \{a_j, b_j, c_j, d_j\}$ is a* 4*-cut in G, and $G_j - S_j$ is a component of $G - S_j$. We say that $\mathcal{A}_j$ determines the* 4*-cut S_j. Note that since $\mathcal{A}_j$ is a component of $\mathcal{K}$, $G_j - \{c_j, d_j\}$ is* 2*-connected. See Figure* 8*.*

LEMMA 3.17. *Let $\mathcal{A}_j$ be a component of $\mathcal{K}$ which determines a* 4*-cut $\{a_j, b_j, c_j, d_j\}$. Then one of the following holds:*

(1) *there exists an induced a_j-b_j path Q in $G_j - \{c_j, d_j\}$ such that $(P - V(P(a_j, b_j))) \cup Q$ is a B_P-augmenting path, or*

(2) *$(G_j, a_j, c_j, b_j, d_j)$ is planar.*

Moreover, one can in $O(|V(G_j)| + |E(G_j)|)$ time find a path as in (1) *or certify that* (2) *holds.*

Proof. Since G is $(4, X_P \cup \{a, b\})$-connected, if $T \subset V(G_j)$ with $|T| \leq 3$, then any component of $G_j - T$ contains an element of $\{a_j, b_j, c_j, d_j\}$. Hence, G_j is $(4, \{a_j, b_j, c_j, d_j\})$-connected. Apply Lemma 2.4 with G_j, a_j, b_j, c_j, d_j as G, a, a', b, b', respectively. Then one of the following holds:

(a) there exists a nonseparating induced a_j-b_j path Q in G_j such that $V(Q) \cap \{c_j, d_j\} = \emptyset$, or

(b) $(G_j, a_j, c_j, b_j, d_j)$ is planar.

Moreover, one can in $O(|V(G_j)| + |E(G_j)|)$ time find a path as in (a) or certify that (b) holds.

If (b) occurs, then we have (2). So we may assume that (a) occurs. Let $P' := (P - V(P(a_j, b_j))) \cup Q$. Then P' is a nonseparating induced path in G. Moreover, since $\{c_j, d_j\}$ is contained in the connected subgraph $G_j - V(Q)$ of $G - V(P')$, $|V(B_P)| < |V(B_{P'})|$. Thus, P' is a B_P-augmenting path. □

We are now ready to prove the main result of this paper. With input G, a, b, P, B_P, and X_P, Algorithm 2 returns a planar a-b chain H in G such that $G - V(H)$ is 2-connected and $V(B_P) \subseteq V(G) - V(H)$.

Algorithm 2. Nonseparating planar chain.

```
Require: $G, a, b, P, B_P, X_P$ satisfying hypotheses of Theorem 1.6.
Return: A planar $a$-$b$ chain $H$ in $G$ such that $G-V(H)$ is 2-connected and $V(B_P) \subseteq V(G) - V(H)$.
 1: loop
 2:   if $G - V(P) = B_P$ then
 3:     Return $H \leftarrow P$ and stop;
 4:   Compute the set $\mathcal{B}$ of $B_P$-bridges in $G - V(P)$;
 5:   Apply Algorithm 1 to $\mathcal{B}$ to compute spanning trees $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_t$ of the components $\mathcal{A}_1, \mathcal{A}_2 \ldots, \mathcal{A}_t$ of the auxiliary graph $\mathcal{K}$, the subpaths $Q_1, Q_2 \ldots, Q_t$ of $P$ and their respective ends $a_1, b_1, a_2, b_2, \ldots, a_t, b_t$;
 6:   if every $\mathcal{A}_j$ determines a 4-cut $\{a_j, b_j, c_j, d_j\}$ and $(G_j, a_j, c_j, b_j, d_j)$ is planar then
 7:     Return $H := (P - \bigcup_{j=1}^{t} V(P(a_j, b_j))) \cup (\bigcup_{j=1}^{t}(G_j - \{c_j, d_j\}))$ and stop;
 8:   if there exists $j$ such that $|R_{\mathcal{A}_j}| \geq 3$ or $\mathcal{A}_j$ does not determine a 4-cut then
 9:     Find a nice bridge $B \in V(\mathcal{A}_j)$;
10:     Find an induced $l_B$-$h_B$ path $Q$ in $G[(V(B) \cup \{l_B, h_B\}]$ such that $(P - V(P(l_B, h_B))) \cup Q$ is a $B_P$-augmenting path;
11:     Set $P \leftarrow (P - V(P(l_B, h_B))) \cup Q$, update $B_P$ and $X_P$, and start a new iteration;
12:   if there exists $\mathcal{A}_j$ which determines a 4-cut and $(G_j, a_j, c_j, b_j, d_j)$ is nonplanar then
13:     Find an induced $a_j$-$b_j$ path $Q$ in $G_j - \{c_j, d_j\}$ such that $(P - V(P(a_j, b_j))) \cup Q$ is a $B_P$-augmenting path.
14:     Set $P \leftarrow (P - V(P(a_j, b_j))) \cup Q$, update $B_P$ and $X_P$, and start a new iteration;
```

Theorem 3.18. *Algorithm* 2 *is correct and runs in* $O(|V(G)||E(G)|)$ *time.*

Proof. Let us first prove the correctness of the algorithm.

At the start of each iteration of the main loop, P is a nonseparating induced a-b path, and B_P is a nontrivial block of $G - V(P)$. Moreover, $G - (V(B_P) - X_P)$ is $(4, X_P \cup \{a, b\})$-connected, where $X_P := N_G(G - V(B_P))$. As the algorithm progresses, $|V(B_P)|$ increases.

If the algorithm stops at line 3, then clearly $G - V(P)$ is 2-connected. Moreover, since P is an induced a-b path, $H := P$ is also a planar a-b chain in G.

If the algorithm stops at line 7, it returns a subgraph H. First, note that $B_P = G - V(H)$ is 2-connected. Let us prove that H is a planar a-b chain in G. Note that $t \geq 1$. For each j, $1 \leq j \leq t$, we have that $|R_{\mathcal{A}_j}| \leq 2$ and $\mathcal{A}_j$ determines a 4-cut $S_j := \{a_j, b_j, c_j, d_j\}$, where $c_j, d_j \in X_P \subseteq V(B_P)$. Moreover, $(G_j, a_j, c_j, b_j, d_j)$ is planar. Since $\mathcal{A}_j$ is a component of $\mathcal{K}$, $G_j - \{c_j, d_j\}$ is 2-connected. Therefore, $H := (P - \bigcup_{j=1}^{t} V(P(a_j, b_j))) \cup (\bigcup_{j=1}^{t}(G_j - \{c_j, d_j\}))$ is a planar a-b chain in G.

If B is a nice bridge, then by Lemma 3.7 the path Q in line 10 exists and $(P - V(P(l_B, h_B))) \cup Q$ is a B_P-augmenting path. So, every time the algorithm executes lines 8–11, it increases $|V(B_P)|$. Moreover, the existence of the nice bridge B on line 9 is guaranteed by Lemma 3.13, (2) of Lemma 3.14, and (2) of Lemma 3.15.

If $\mathcal{A}_j$ is a component of $\mathcal{K}$ that determines a 4-cut $\{a_j, b_j, c_j, d_j\}$ and $(G_j, a_j, c_j, b_j, d_j)$ is nonplanar, then by (1) of Lemma 3.17 the path Q in line 13 exists and $(P - V(P(a_j, b_j))) \cup Q$ is a B_P-augmenting path. So when the algorithm executes lines 12–14, it also increases $|V(B_P)|$.

Finally, Lemma 3.2 guarantees that after the update of B_P and X_P either in line 11 or in line 14, the hypotheses of Theorem 1.6 still hold in the next iteration. Since $|V(B_P)|$ increases at each iteration, the loop eventually stops, and hence Algorithm 2 is correct.

Now, let us verify the complexity of the algorithm.

The loop on line 1 is executed at most $|V(G)|$ times since $|V(B_P)|$ increases at each iteration.

The steps in lines 2 and 4 can be performed in $O(|V(G)|+|E(G)|)$ time by standard graph search techniques (for example, see [18]). By Lemma 3.12, the spanning trees $\mathcal{T}_1,\dots,\mathcal{T}_t$ (line 5), the paths $Q_1,\dots,Q_t$, and their respective ends $a_1,b_1,\dots,a_t,b_t$ can be computed in $O(|V(G)|+|E(G)|)$ time by Algorithm 1.

The steps in line 6 and line 12 test whether (G_j,a_j,b_j,c_j,d_j) is planar. By Lemma 2.4 this is equivalent to deciding whether there exists a nonseparating induced a_j-b_j path in G_j containing neither c_j nor d_j and can be done in $O(|V(G_j)|+|E(G_j)|)$ (and hence $O(|V(G)|+|E(G)|)$ time).

Finding a nice bridge B in line 9 can be done in $O(|V(G)|)$ time by Lemma 3.13, (2) of Lemma 3.14, and (2) of Lemma 3.15. The path Q in line 10 can be found in $O(|V(G)|+|E(G)|)$ time by Lemma 3.7. The path Q in line 13 can be found in $O(|V(G_j)|+|E(G_j)|)$ (and hence $O(|V(G)|+|E(G)|)$ time by Lemma 3.17.

Clearly, the steps in lines 11 and 14 can be done in $O(|E(G)|)$ time.

Therefore, the running time of Algorithm 2 is $O(|V(G)||E(G)|)$. □

4. Related results. Our eventual goal is to construct a decomposition of any 4-connected graph into certain chains and find four independent spanning trees. This will be done in forthcoming papers where the asymptotic performance of Algorithm 2 can often be improved to $O(|V(G)|^2)$: instead of applying the algorithm to G, we apply it to a sparse spanning 4-connected subgraph of G with the help from a result of Ibaraki and Nagamochi [9].

THEOREM 4.1. *Given a k-connected graph G, one can find in $O(|V(G)|+|E(G)|)$ time a spanning k-connected subgraph of G with at most $k|V(G)|$ edges.*

The first step in our decomposition of a 4-connected graph is to find a nonseparating cyclic chain. Intuitively, a cyclic chain is a graph obtained from a chain by identifying its ends. More precisely, we have the following.

DEFINITION 4.2. *A connected graph H is a* cyclic chain *if for some integer $k \geq 2$, there exist subgraphs $B_1,\dots,B_k$ of H and vertices $v_1,\dots,v_k$ of H such that*

(i) *for $1 \leq i \leq k$, B_i is 2-connected or B_i is induced by an edge of H,*

(ii) *$V(H)=\bigcup_{i=1}^k V(B_i)$ and $E(H)=\bigcup_{i=1}^k E(B_i)$,*

(iii) *if $k=2$, then $V(B_1)\cap V(B_2)=\{v_1,v_2\}$ and $E(B_1)\cap E(B_2)=\emptyset$, and*

(iv) *if $k\geq 3$, then $V(B_i)\cap V(B_{i+1})=\{v_i\}$ for $1\leq i\leq k$, where $B_{k+1}:=B_1$, and $V(B_i)\cap V(B_j)=\emptyset$ for $1\leq i<i+2\leq j\leq k$ and $(i,j)\neq(1,k)$.*

We usually fix one of the vertices $v_1,\dots,v_k$ as the root *of H, say, v_k, and we use the notation $H:=v_0B_1v_1\dots v_{k-1}B_kv_k$ to indicate that H is a* cyclic chain rooted at v_0 *$(=v_k)$. Each subgraph B_i is called a* piece *of H. See Figure 9 for an example with $k=6$.*

DEFINITION 4.3. *Let G be a graph and let $H:=v_0B_1v_1\dots v_{k-1}B_kv_k$ be a cyclic chain rooted at $v_0=v_k$. If H is an induced subgraph of G, then we say that H is a* cyclic chain in G. *We say that H is a* planar cyclic chain in G *if for each $1\leq i\leq k$ with $|V(B_i)|\geq 3$ (or equivalently, B_i is 2-connected), there exist distinct vertices $x_i,y_i\in V(G)-V(H)$ such that $(G[V(B_i)\cup\{x_i,y_i\}]-x_iy_i,x_i,v_{i-1},y_i,v_i)$ is planar, and $B_i-\{v_{i-1},v_i\}$ is a component of $G-\{x_i,y_i,v_{i-1},v_i\}$. See Figure 10 for an*

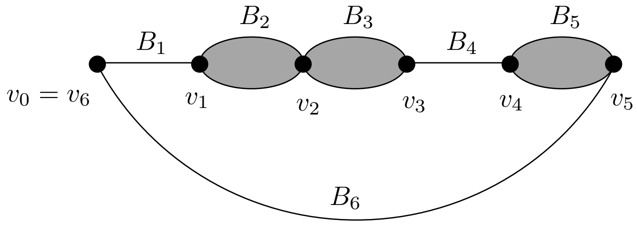


FIG. 9. *Example of a cyclic chain.*

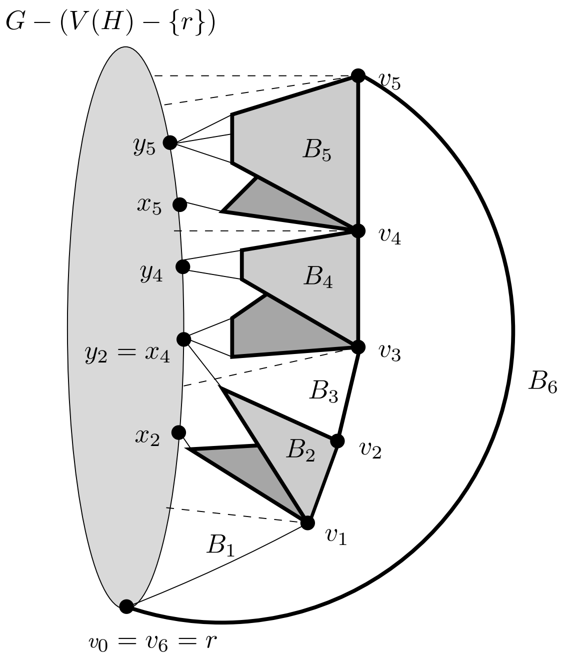


FIG. 10. *A planar cyclic chain $H := v_0B_1v_1 \ldots v_5B_6v_6$ rooted at r in a graph G.*

example with $k = 6$, where the dashed edges may or may not exist in G but they are not part of H.

Now we can state and prove a result which will serve as the first chain in a chain decomposition of a 4-connected graph. See again Figure 10 for an example.

THEOREM 4.4. *Let G be a 4-connected graph and let $ra \in E(G)$. Then there exists a planar cyclic chain H in G rooted at r such that ra induces a piece of H and $G - (V(H) - \{r\})$ is 2-connected. Moreover, such a chain can be found in $O(|V(G)||E(G)|)$ time.*

Proof. Let G be a 4-connected graph and let $ra \in E(G)$. By Theorem 1.2, one can find a nonseparating induced cycle C in G through ra in $O(|V(G)| + |E(G)|)$ time. Let P denote the path $C - r$ and let b be the end of P other than a. Since C is induced, exactly two neighbors of r lie on P, namely, a and b. Thus, since $(G - V(P)) - r = G - V(C)$ is connected, r is not a cut vertex of $G - V(P)$. Let B_P be the block of $G - V(P)$ containing r. Note that $N_G(r) \subseteq V(B_P) \cup \{a, b\}$. Hence, B_P contains more than two vertices because r has degree at least four, and therefore, B_P is 2-connected. If $G - V(P) = B_P$, then $H := C$ is a planar cyclic chain rooted at

r such that ra induces a piece of H and $G-(V(H)-\{r\})$ is 2-connected. So assume that $G-V(P)$ is not 2-connected, that is, $G-V(P)\neq B_P$.

Let $X_P := N_G(G-V(B_P))$. Then G,a,b,P,B_P,X_P satisfy the hypotheses of Theorem 1.6. By Theorem 1.6 one can find in $O(|V(G)||E(G)|)$ time a planar a-b chain H' in G such that $B_P\subseteq G-V(H')$ and $G-V(H')$ is 2-connected. Since $N_G(r)\subseteq V(B_P)\cup\{a,b\}$, we have $r\notin N_G(H'-\{a,b\})$. Therefore, $H:=(H'\cup\{r\})+\{ra,rb\}$ is an induced subgraph of G. Hence H is a planar cyclic chain in G rooted at r such that ra induces a piece of H and $G-(V(H)-\{r\})$ is 2-connected. □

The property that ra induces a piece in the planar cyclic chain in Theorem 4.4 is not necessary for constructing a chain decomposition of a 4-connected graph, but it has an interesting consequence (see Corollary 1.8). To derive Corollaries 1.7 and 1.8, we need to introduce some results on Hamilton paths and cycles in planar graphs.

Thomassen [20] proved the existence of a special path in a 2-connected planar graph, and, later on, Chiba and Nishizeki [4] developed a $O(|V(G)|+|E(G)|)$ algorithm for finding such a path.

THEOREM 4.5. *Let G be a 2-connected plane graph with a facial cycle F. Let $x\in V(F)$, $e\in E(F)$, and $y\in V(G)-\{x\}$. Then G contains an x-y path P through e such that*

(i) *every P-bridge of G has at most three attachments on P, and*

(ii) *every P-bridge of G containing an edge of F has at most two attachments on P.*

Moreover, such a path P can be found in $O(|V(G)|+|E(G)|)$ time.

For our purpose, we need the following consequence of Theorem 4.5. This was proved by Curran and Yu [5]. For a proof, see the appendix.

COROLLARY 4.6. *Let (G,a,c,b,d) be a planar graph and suppose that G is $(4,\{a,b,c,d\})$-connected. Then there exists a Hamilton a-b path in $G-\{c,d\}$. Moreover, such a path can be found in $O(|V(G)|+|E(G)|)$ time.*

COROLLARY 4.7. *Let G be a 4-connected graph and let H be a planar x-y chain in G. Then there exists a Hamilton x-y path in H. Moreover, such a path can be found in $O(|V(H)|+|E(H)|)$ time.*

Proof. Let $H:=v_0B_1v_1\dots v_{k-1}B_kv_k$, where $v_0=x$ and $v_k=y$. Since H is a planar chain, for each nontrivial block B_i of H there exists $u_i,w_i\in V(G)-V(H)$ such that $(G[V(B_i)\cup\{u_i,w_i\}]-u_iw_i,v_{i-1},u_i,v_i,w_i)$ is planar and $B_i-\{v_{i-1},v_i\}$ is a component of $G-\{v_{i-1},v_i,u_i,w_i\}$. Moreover, $G_i:=G[V(B_i)\cup\{u_i,w_i\}]-u_iw_i$ is $(4,\{v_{i-1},v_i,u_i,w_i\})$-connected. Applying Corollary 4.6 to $(G_i,v_{i-1},u_i,v_i,w_i)$ as (G,a,c,b,d), one can find a Hamilton v_{i-1}-v_i path in $B_i=G_i-\{u_i,w_i\}$ in $O(|V(G_i)|+|E(G_i)|)$ time. Therefore, a Hamilton x-y path in H can be found in $O(|V(H)|+|E(H)|)$ time. □

By an argument similar to that in Corollary 4.7 we can prove the following.

COROLLARY 4.8. *Let G be a 4-connected graph and let H be a planar cyclic chain in G. Then there exists a Hamilton cycle in H. Moreover, such a cycle can be found in $O(|V(H)|+|E(H)|)$ time.*

It is now easy to see that Corollary 1.7 follows from Theorem 1.6 and Corollary 4.7, and Corollary 1.8 follows from Theorems 4.1 and 4.4 and Corollary 4.8.

Corollary 1.8 is similar in spirit to Theorem 1.2, which was proved by Tutte. Unlike Tutte's result, however, we cannot ask the cycle C in Corollary 1.8 to be induced, and we do not remove the vertex r from the graph. Curran and Yu [5, Theorem 1.3] showed that if G is 5-connected and $e\in E(G)$, then G contains an induced cycle C through e such that $G-V(C)$ is 2-connected. All these results are related to the following important open problem. In 1975, Lovász [13] conjectured the following. Given any positive integer k, there exists some positive integer $f(k)$ with the property that for any given vertices x and y of an $f(k)$-connected graph G, there exists an induced x-y path P in G such that $G - V(P)$ is k-connected. Thus, Tutte's result solves the case $k = 1$, and Curran and Yu's result implies the case $k = 2$, which was proved independently by Chen, Gould, and Yu [1] and Kriesell [12]. The conjecture is still open for higher values of k.

Appendix.

Proof of Lemma 2.2. First, we prove that exactly one of (1) and (2) holds. Clearly, (1) and (2) are mutually exclusive because of planarity. We know that either (1) or (2) of Theorem 2.1 holds. If (1) of Theorem 2.1 holds, then (1) of Lemma 2.2 holds. So assume (2) of Theorem 2.1 holds. Let $A_1, \dots, A_k$ be as described in (2) of Theorem 2.1. Then $S \cap A_i = \emptyset$ for $1 \leq i \leq k$. Hence, $G[A_i]$ consists of those components of $G - N_G(A_i)$ containing no element of S, contradicting our assumption that G is $(4, S)$-connected because $|N_G(A_i)| \leq 3$. Thus no A_i can exist. Let G' be described as in (2) of Theorem 2.1. Observe that $(G' - \{ab, cd\}, a, c, b, d)$ is planar. Since $G' - \{ab, cd\} = G$, (2) of Lemma 2.2 holds. Therefore, either (1) or (2) holds.

Let us prove the algorithmic part of Lemma 2.2. First, we give a sketch of Shiloach's algorithm. It has as input a graph G and vertices a, b, c, d of G (with no connectivity hypothesis on G). The algorithm consists of reductions R1, . . . , R6, which reduces the general problem to a restricted one.

R1: The algorithm initially reduces the problem to 3-connected graphs, in $O(|V(G)| + |E(G)|)$ time.

R2: If the graph is planar, then a specialized $O(|V(G)| + |E(G)|)$ algorithm for 3-connected planar graphs [15] is used to solve the problem.

R3: Assume that G is nonplanar. This is the most time-consuming step of the algorithm. It reduces the problem using network flow techniques to graphs satisfying some connectivity constraints involving $S := \{a, b, c, d\}$. Namely, the resulting graph G satisfies the following property: for any subset S' of vertices of G with $|S'| \leq 4$, there exist four disjoint paths connecting S to S' (these paths can share ends in S', however). In fact, this step is not executed at once, but it is interspersed with reductions R4, R5, and R6 in the algorithm. Whenever the algorithm finds a set S' which is not connected to S by four disjoint paths, a reduction is performed. The total time spent with these reductions during the whole algorithm is $O(|V(G)||E(G)|)$. For simplicity, suppose that no such set S' exists. By Menger's theorem, this is equivalent to saying that G is $(4, S)$-connected. Note that the graph G in the statement of Lemma 2.2 is $(4, S)$-connected.

Thus, so far G is 3-connected, nonplanar, and $(4, S)$-connected. The algorithm then finds a subdivision of a Kuratowski graph (K_5 or $K_{3,3}$). Shiloach gave an $O(|V(G)|^2)$ algorithm to find such a subdivision, but this can be improved as we show below using an algorithm of Hsu and Shih [7].

R4: If a subdivision of K_5 is found, Shiloach claims that the required two disjoint paths can be found in $O(|V(G)| + |E(G)|)$ time, using a result of Watkins [22].

R5 and R6: If a subdivision of $K_{3,3}$ is found, then Shiloach's algorithm finds the required two disjoint paths in $O(|V(G)| + |E(G)|)$ time.

Let us show how to improve the running time of the algorithm for $(4, S)$-connected graphs. Let G be a graph, let $S := \{a, b, c, d\} \subset V(G)$ and suppose that G is $(4, S)$-connected. Let $G^+ := G + \{ac, cb, bd, da\}$. Since G is $(4, S)$-connected, G^+ is $(4, S)$-connected. Because each of a, b, c, d has degree at least three in G^+, it follows that

G^+ is 3-connected. Moreover, if there exist disjoint paths P and Q in G^+ from a to b, and from c to d, respectively, then P and Q are both paths in G, and vice versa.

We describe now how to solve the two disjoint paths problem for G^+ in $O(|V(G)|+|E(G)|)$ time. Hsu and Shih [7] developed a $O(|V(H)|+|E(H)|)$ algorithm that, given a graph H, either finds an embedding of H or finds a subdivision of a Kuratowski graph in H. Applying this algorithm to G^+, we find either an embedding of G^+ or a subdivision of K_5 or $K_{3,3}$. If the former occurs, then G^+ is planar, and we can use step R2 to solve the two disjoint paths problem in $O(|V(G)| + |E(G)|)$ time. Otherwise, there exists a subdivision of K_5 (or $K_{3,3}$), and we can use steps R4 (or R5 and R6, respectively) of Shiloach's algorithm to find the required two disjoint paths. Thus, we can find the two disjoint paths P and Q, if they exist, in $O(|V(G)| + |E(G)|)$ time. □

Proof of Corollary 4.6. Let $G' := (G - d) \cup \{bc, ac\}$. We first show that G' is 2-connected. Suppose on the contrary that G' is not 2-connected. Let x be a cut vertex of G'. Since G is $(4, \{a, b, c, d\})$-connected, $G - \{c, d\}$ contains an a-b path, and hence $\{a, b, c\}$ is contained in a cycle in G'. Therefore, $\{a, b, c\}$ is contained in an x-bridge of G', and G' has another x-bridge B such that $(V(B) - \{x\}) \cap \{a, b, c\} = \emptyset$. Hence, $B - x$ is a component of $G - T$, where $T := \{x, d\}$, and $(V(B) - \{x\}) \cap \{a, b, c\} = \emptyset$, a contradiction.

Note that G' is planar and can be drawn in the plane so that ac, bc and $N_G(d)$ are on a facial cycle F. Applying Theorem 4.5 (with G', a, c, bc as G, x, y, e, respectively), G' has an a-c path P through bc satisfying (i) and (ii) of Theorem 4.5. Moreover, such a path can be found in $O(|V(G)| + |E(G)|)$ time. Note that $ac \notin E(P)$ because $bc \in E(P)$.

We proceed to show that every P-bridge of G' is induced by a single edge, and so P must be a Hamilton path in G'. Let B be a P-bridge of G' such that $V(B) - V(P) \neq \emptyset$, and let $T := V(B) \cap V(P)$. Since a, b and c are all on P, then $\{a, b, c\} \cap V(B) \subseteq T$. Thus, $B - T$ is a component of $G - (T \cup \{d\})$ containing no element of $\{a, b, c, d\}$. If $|T| \leq 2$, then $|T \cup \{d\}| \leq 3$, contradicting our assumption that G is $(4, \{a, b, c, d\})$-connected. Since P satisfies (i) of Theorem 4.5, we may assume $|T| = 3$. Then by (ii) of Theorem 4.5, $E(B) \cap E(F) = \emptyset$, and hence $(V(B) - T) \cap N_G(d) = \emptyset$. Therefore, $B - T$ is a component of $G - T$ such that $(V(B) - T) \cap \{a, b, c, d\} = \emptyset$, a contradiction to the assumption that G is $(4, \{a, b, c, d\})$-connected.

Thus, $P - c$ is a Hamilton a-b path in $G - \{c, d\}$. Moreover, by Theorem 4.5 such a path can be found in $O(|V(G)| + |E(G)|)$ time. □

REFERENCES

[1] G. Chen, R. Gould, and X. Yu, *Graph connectivity after path removal*, Combinatorica, 23 (2003), pp. 185–203.

[2] K. Chakravarti and N. Robertson, *Covering three edges with a bond in a non-separable graph*, Ann. Discrete Math., 8 (1979), p. 247.

[3] J. Cheriyan and S. N. Maheshwari, *Finding nonseparating induced cycles and independent spanning trees in* 3*-connected graphs*, J. Algorithms, 9 (1988), pp. 507–537.

[4] N. Chiba and T. Nishizeki, *The Hamiltonian cycle problem is linear-time solvable for* 4*-connected planar graphs*, J. Algorithms, 10 (1989), pp. 187–211.

[5] S. Curran and X. Yu, *Nonseparating cycles in* 4*-connected graphs*, SIAM J. Discrete Math., 16 (2003), pp. 616–629.

[6] D. Dolev, J. Y. Halpern, B. Simons, and R. Strong, *A new look at fault tolerant network routing*, in Proceedings of the 16th Annual ACM Symposium on Theory of Computing, 1984, pp. 526–535.

[7] W.-L. Hsu and W.-K. Shih, *A new planarity test*, Theoret. Comput. Sci., 223 (1999), pp. 179–191.

[8] A. Huck, *Independent trees in graphs*, Graphs Combin., 10 (1994), pp. 29–45.

[9] T. Ibaraki and H. Nagamochi, *A linear-time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph*, Algorithmica, 7 (1992), pp. 583–596.

[10] A. Itai and M. Rodeh, *The multi-tree approach to reliability in distributed networks*, in Proceedings of the 25th Annual IEEE Symposium on Foundations of Computer Science 1984, pp. 137–147.

[11] A. Itai and A. Zehavi, *Three tree-paths*, J. Graph Theory, 13 (1989), pp. 175–188.

[12] M. Kriesell, *Induced paths in* 5*-connected graphs*, J. Graph Theory, 36 (2001), pp. 52–58.

[13] L. Lovász, *Problems,* in Recent Advances in Graph Theory, M. Fiedler, ed., Academia, Prague, 1975, pp. 541–544.

[14] K. Miura, S. Nakano, T. Nishizeki, and D. Takahashi, *A linear-time algorithm to find four independent spanning trees in four connected planar graphs*, Internat. J. Found. Comput. Sci., 10 (1999), pp. 195–210.

[15] Y. Perl and Y. Shiloach, *Finding two disjoint paths between pairs of vertices in a graph*, J. Assoc. Comput. Mach., 25 (1978), pp. 1–9.

[16] P. D. Seymour, *Disjoint paths in planar graphs*, Discrete Math., 29 (1980), pp. 293–309.

[17] Y. Shiloach, *A polynomial solution to the undirected two paths problem*, J. Assoc. Comput. Mach., 27 (1980), pp. 445–456.

[18] R. E. Tarjan, *Data Structures and Network Algorithms*, SIAM, Philadelphia, 1983.

[19] C. Thomassen, 2*-linked graphs*, European J. Combin., 1 (1980), pp. 371–378.

[20] C. Thomassen, *A theorem on paths in planar graphs*, J. Graph Theory, 7 (1983), pp. 169–176.

[21] W. T. Tutte, *How to draw a graph*, Proc. London Math Soc., 13 (1963), pp. 743–767.

[22] M. Watkins, *On the existence of certain disjoint arcs in graphs*, Duke Math. J., 35 (1968), pp. 321–346.

SIAM J. Discrete Math.
Vol. 19, No. 2, pp. 420–430

RESOLVABLE STEINER QUADRUPLE SYSTEMS FOR THE LAST 23 ORDERS[*]

L. JI[†] AND L. ZHU[†]

Abstract. A Steiner quadruple system of order v, denoted by SQS(v), is a pair $(X, \mathcal{B})$, where X is a v-set of points and $\mathcal{B}$ is a set of 4-subsets of X, called blocks, with the property that every 3-subset of X is contained in exactly one block of $\mathcal{B}$. An SQS(v) is resolvable if its block set can be partitioned into parallel classes. In 1987, Hartman showed that the necessary condition $v \equiv 4$ or 8 (mod 12) for the existence of a resolvable SQS(v) is sufficient with 23 possible exceptions. In this paper, we construct resolvable SQS(v)s for the 23 undecided orders.

Key words. resolvable Steiner quadruple system, candelabra quadruple system, H design

AMS subject classification. 05B05

DOI. 10.1137/S0895480104440584

1. Introduction.

A *Steiner quadruple system* of order v, denoted by SQS(v), is a pair $(X, \mathcal{B})$ where X is a v-set of *points* and $\mathcal{B}$ is a set of 4-subsets of X, called *blocks*, with the property that every 3-subset of X is contained in exactly one block of $\mathcal{B}$. Hanani [4] showed that an SQS(v) exists if and only if $v \equiv 2$ or 4 (mod 6).

If $(X, \mathcal{B})$ is an SQS(v), then $P \subset \mathcal{B}$ is a *parallel class* if P is itself a partition of X. We say $(X, \mathcal{B})$ is *resolvable* (and denoted it by RSQS(v)) if $\mathcal{B}$ can be partitioned into $r(v) = \frac{1}{3}\binom{v-1}{2}$ parts $\mathcal{B} = P_1 \cup P_2 \cup \cdots \cup P_{r(v)}$, such that each part P_i is a parallel class. In this case we shall say that $P_1 \mid P_2 \mid \cdots \mid P_{r(v)}$ is a *resolution* of $\mathcal{B}$. It is clear that a necessary condition for the existence of an RSQS(v) is that $v \equiv 4$ or 8 (mod 12) or $v = 1$ or 2.

In 1977, the only orders for which an RSQS(v) was known were $v = 2^n$, and the only recursive construction known was the doubling construction (i.e., a construction of an RSQS($2v$) from an RSQS(v)). In 1978, Booth [1] and Greenwell and Lindner [2] constructed an RSQS(20) and an RSQS(28), thus providing the first examples with v not a power of two. Further examples were given by Hartman, who constructed RSQS($q+1$) for all prime powers $q \equiv 7$ (mod 12) with $q \le 379$ [5] and RSQS($4p$) for $p \in \{19, 43, 127, 199, 223, 271, 1603\}$ [6].

Hartman [5], [7] provided the main recursive theorems for RSQS(v)s. To state this result, we need the notion of a resolvable quadruple system with a resolvable subsystem. Let $(X, \mathcal{B})$ be an RSQS(v) with a resolution $P_1 \mid P_2 \mid \cdots \mid P_{r(v)}$ and let $(Y, \mathcal{A})$ be an RSQS(u) with a resolution $P'_1 \mid P'_2 \mid \cdots \mid P'_{r(u)}$ such that $Y \subset X$, $\mathcal{A} \subset \mathcal{B}$ and $P'_j \subset P_j$ for $1 \le j \le r(u)$. Then $(X, \mathcal{B})$ is an RSQS(v) *with a resolvable subsystem* of order u and will be denoted by RSQS($v : u$). Note that subsystems of orders 1 and 2 are trivially resolvable.

Theorem 1.1 (see [7]). *If there exists an RSQS*($g+s : s$) *where* $g \equiv s$ (*mod* 12)*, then there exists an RSQS*($3g+s : g+s$).

Theorem 1.2 (see [5]). *If there exists an RSQS*($g+s : s$) *where* $g \equiv 0$ (*mod* 12) *and* $9s \ge 5g$*, then there exists an RSQS*($3g+s : g+s$).

[*] Received by the editors February 4, 2004; accepted for publication (in revised form) December 24, 2004; published electronically October 31, 2005.
http://www.siam.org/journals/sidma/19-2/44058.hmtl

[†] Department of Mathematics, Suzhou University, Suzhou 215006, China (jilijun@suda.edu.cn).

In 1987, Hartman used these two recursive constructions and some small designs to obtain the following theorem.

THEOREM 1.3 (see [6]). *There exists an RSQS*(v) *for all* $v \equiv 4$ *or* 8 (mod 12) *with* 23 *possible exceptions* $v \in \{220, 236, 292, 364, 460, 596, 676, 724, 1076, 1100, 1252, 1316, 1820, 2236, 2308, 2324, 2380, 2540, 2740, 2812, 3620, 3820, 6356\}$.

In this paper, we shall construct RSQSs of the above 23 orders. Thus, Theorem 1.3 can be updated in the following.

THEOREM 1.4. *If* v *is a positive integer with* $v > 2$, *then an RSQS*(v) *exists if and only if* $v \equiv 4$ *or* 8 (mod 12).

In section 2, we introduce a resolvable $H(u, g, 4, 3)$ and use it to construct RSQSs. In section 3, we introduce a resolvable candelabra quadruple system and use it to give recursive constructions for RSQSs. Theorem 1.4 will be proved in section 4.

2. Constructions using resolvable *H* designs. Mills [9] discussed the existence of H designs. Let u be a positive integer and let X be a set of points. Let $\mathcal{T} = \{T_1, \ldots, T_u\}$ be a partition of X into disjoint sets T_i, which we will call the groups of $\mathcal{T}$. By a transverse of $\mathcal{T}$ we mean a subset of X that meets each T_i in at most one point. An H design is a triple $(X, \mathcal{T}, \mathcal{B})$, where $\mathcal{B}$ is a set of k-element transverses of $\mathcal{T}$, called blocks, such that each t-element transverse is contained in exactly one of them. If $|T_1| = \cdots = |T_u| = g$, we denote it by $H(u, g, k, t)$. Mills [9] showed that for $u > 3, u \neq 5$, an $H(u, g, 4, 3)$ exists if and only if ug is even and $g(u-1)(u-2)$ is divisible by 3 and that for $u = 5$, an $H(5, g, 4, 3)$ exists if g is divisible by 4 or 6.

An $H(u, g, 4, 3)$ is said to be *resolvable* if its block set can be partitioned into $(u-1)(u-2)g^2/6$ parts P_i such that each P_i is a partition of the point set (called parallel class). It is easy to see that if a resolvable $H(u, g, 4, 3)$ exists, then $ug \equiv 0 \pmod 4$.

For $u = 4$, we have the following result on resolvable $H(4, g, 4, 3)$.

LEMMA 2.1. *There is a resolvable* $H(4, g, 4, 3)$ *for any positive integer* g.

Proof. The desired design is based on $Z_4 \times Z_g$ with groups $\{i\} \times Z_g$, $0 \leq i \leq 3$. Let $\mathcal{F} = \{F_1, \ldots, F_g\}$ and $\mathcal{F}' = \{F'_1, \ldots, F'_g\}$ be a one-factorization of complete bipartite graph on $\{0, 1\} \times Z_g$ and $\{2, 3\} \times Z_g$ with groups as partites, respectively. For any $\{x, y\} \in F_k$ and $\{x', y'\} \in F'_k$, construct blocks $\{x, y, x', y'\}$, where $1 \leq k \leq g$. Denote the set of all these blocks by $\mathcal{C}$. It is easy to see that $\mathcal{C}$ is the block set of an $H(4, g, 4, 3)$. We need only to give its required g^2 parallel classes $P(i, j)$, $1 \leq i, j \leq g$. Each parallel class $P(i, j)$ comprises all blocks $\{x, y, x', y'\}$, where $\{x, y\}$ is the kth edge of F_i and $\{x', y'\}$ is the $(k+j)$th edge of F'_i. It is easy to check that all $P(i, j)$ form a partition of $\mathcal{C}$. It follows that a resolvable $H(4, g, 4, 3)$ exists. □

LEMMA 2.2. *Suppose that there exists a resolvable* $H(u, g, 4, 3)$ *where* u *is even. If there is an RSQS*($2gq$), *then there is an RSQS*(ugq).

Proof. Let $(X, \mathcal{G}, \mathcal{B})$ be the given resolvable $H(u, g, 4, 3)$ with a resolution $P_1 \mid P_2 \mid \cdots \mid P_s$, where $s = (u-1)(u-2)g^2/6$ and $\mathcal{G} = \{G_0, \ldots, G_{u-1}\}$. Let $\mathcal{F}' = \{F'_1, \ldots, F'_{u-1}\}$ be a one-factorization of complete graph on Z_u since u is even.

First, we construct a resolvable $H(u, gq, 4, 3)$ on $X' = X \times Z_q$ with the group set $\mathcal{G}' = \{G'_i = G_i \times Z_q : 0 \leq i \leq u-1\}$.

For each block $B \in \mathcal{B}$, construct a resolvable $H(4, q, 4, 3)$ on $B \times Z_q$ with groups $\{x\} \times Z_q$, $x \in B$. Such a design exists by Lemma 2.1. Denote its block set by $\mathcal{A}_B$ and its parallel classes by $P_B(r, h)$, $1 \leq r, h \leq q$.

Let $\mathcal{A} = \cup_{B \in \mathcal{B}} \mathcal{A}_B$. Then $(X', \mathcal{G}', \mathcal{A})$ is an $H(u, gq, 4, 3)$ and its block set $\mathcal{A}$ can be partitioned into $(u-1)(u-2)g^2q^2/6$ parallel classes $P(j, r, h)$, $1 \leq j \leq s$, $1 \leq r, h \leq q$, where $P(j, r, h) = \cup_{B \in P_j} P_B(r, h)$. So, the resultant $H(u, gq, 4, 3)$ is also resolvable.

Now, we shall convert the above $H(u, gq, 4, 3)$ to an RSQS(ugq).

Since an RSQS$(2gq)$ exists, gq is even. For $0 \leq i \leq u-1$, let $\mathcal{F}^i = \{F_1^i, \ldots, F_{gq-1}^i\}$ be a one-factorization of complete graph on G_i'. For any $\{x, y\} \in F_l^a$ and $\{x', y'\} \in F_l^b$, construct blocks $\{x, y, x', y'\}$, where $1 \leq l \leq gq-1$ and $\{a, b\} \in F_i'$, $2 \leq i \leq u-1$. Denote the set of all these blocks by $\mathcal{A}'$.

For $1 \leq k \leq u/2$, let the kth edge of F_1' be $\{a, b\}$. Construct an RSQS$(2gq)$ on $G_a' \cup G_b'$. Such a design exists by assumption. Denote its block set by $\mathcal{C}_k$ and its parallel classes by $Q(k, 1), \ldots, Q(k, (2gq-1)(2gq-2)/6)$. Let $\mathcal{C} = \cup_{1 \leq k \leq u/2} \mathcal{C}_k$.

Then it is easy to check that $(X', \mathcal{G}', \mathcal{A} \cup \mathcal{A}' \cup \mathcal{C})$ is an SQS(ugq). We need to give its required parallel classes.

The first $(u-1)(u-2)g^2q^2/6$ parallel classes are $P(j, r, h)$. For $2 \leq i \leq u-1$, $1 \leq l \leq gq-1$ and $1 \leq m \leq gq/2$, each parallel class $P'(i, l, m)$ comprises all blocks $\{x, y, x', y'\}$, where $\{x, y\}$ is the kth edge of F_l^a, $\{x', y'\}$ the $(m+k)$th edge of F_l^b, and $\{a, b\} \in F_i'$. Note that all $P'(i, l, m)$s form a partition of $\mathcal{A}'$. For $1 \leq j' \leq (2gq-1)(2gq-2)/6$, let $Q'(j') = \cup_{1 \leq k \leq u/2} Q(k, j')$. Then $Q'(j')$ is a parallel class on X' and all $Q'(j')$s form a partition of $\mathcal{C}$. So, $Q'(j')$, $P(j, r, h)$ and $P'(i, l, m)$ are pairwise disjoint and form a resolution of the block set of the resultant SQS. The result then follows. □

The above lemma indicates that resolvable H designs are useful in the construction of RSQSs. A construction of resolvable H designs is given below.

To construct a resolvable $H(v, 2, 4, 3)$ we shall modify the well-known doubling construction. Let $(V, \mathcal{B})$ be an SQS(v). Let $\{F_1, \ldots, F_{v-1}\}$ be a one-factorization of K_v defined on the vertex set V. The doubling construction produces an SQS$(2v)$ $(V \times Z_2, \mathcal{B}_1 \cup \mathcal{B}_2)$, where

$$\mathcal{B}_1 = \{B \times \{i\} : B \in \mathcal{B}, i \in Z_2\},$$

$$\mathcal{B}_2 = \{(E \times \{0\}) \cup (E' \times \{1\}) : E, E' \in F_j, 1 \leq j \leq v-1\}.$$

We first remark that the block set $\mathcal{B}_2$ can be partitioned into parallel classes. For convenience, let

$$F_j = \{\{a_{0j}, b_{0j}\}, \{a_{1j}, b_{1j}\}, \ldots, \{a_{(n-1)j}, b_{(n-1)j}\}\},$$

where $v = 2n$. For $1 \leq j \leq v-1$ and for $0 \leq k \leq n-1$, denote

$$\mathcal{B}(j, k) = \{(\{a_{ij}, b_{ij}\} \times \{0\}) \cup (\{a_{(i+k)j}, b_{(i+k)j}\} \times \{1\}) : 0 \leq i \leq n-1\},$$

where $i + k$ is reduced modulo n. It is clear that $\mathcal{B}(j, k)$ is a parallel class and also

$$\mathcal{B}_2 = \bigcup_{1 \leq j \leq v-1} \bigcup_{0 \leq k \leq n-1} \mathcal{B}(j, k).$$

We further remark that we can obtain an $H(v, 2, 4, 3)$ based on $V \times Z_2$ with groups $\{x\} \times Z_2, x \in V$. Let $\mathcal{D} = \cup_{1 \leq j \leq v-1} \mathcal{B}(j, 0)$ and let $\mathcal{B}_3 = \mathcal{B}_2 \setminus \mathcal{D}$. It is easy to see that $\mathcal{B}_1 \cup \mathcal{B}_3$ is the block set of the H design. What we want is a resolvable $H(v, 2, 4, 3)$. Since $\mathcal{B}_3$ is the union of parallel classes, we shall pick up some of these parallel classes together with $\mathcal{B}_1$ and reorganize them into new parallel classes. To do this we require some structure on both $\mathcal{B}_1$ and F_j; here we let $V = Z_{2n}$ and a group $G = (Z_{2n}, +)$ will be used.

Let R be the subgroup of order n in G, so

$$R = \{0, 2, 4, \ldots, 2n-2\}.$$

Let K_i $(i = 0, 1)$ be a graph on the vertex set Z_{2n} such that $\{a, b\}$ is an edge if and only if $a-b$ and i have the same parity. Then K_0 is an $(n-1)$-regular graph and K_1 is an n-regular graph, $K_0 \cup K_1 = K_{2n}$.

For $\Delta \subset Z_{2n}$, let $\Delta + r = \{\delta + r : \delta \in \Delta\}$. Denote

$$dev(\Delta) = \{\Delta + r : r \in R\}.$$

For odd $d = 1, 3, \ldots, 2n-1$, it is obvious that $dev(\{0, d\})$ is a one-factor of K_{2n}, denoted by $\Gamma(d)$, and K_1 is decomposed into n one-factors $\Gamma(d)$ for odd $d, 1 \leq d \leq 2n-1$. Although there may be no such a one-factorization for K_0, we can show that $K_0 \cup \Gamma(d)$ has a one-factorization under certain conditions.

Lemma 2.3. *Let n be a prime > 3. For any odd d, $1 \leq d \leq 2n-1$, the n-regular graph $K_0 \cup \Gamma(d)$ has a one-factorization.*

Proof. For $d = 1$, we have a one-factor F of $K_0 \cup \Gamma(1)$:

$$\{0,1\}, \{-2,2\}, \{-1,3\}, \ldots, \{-2i, 2i\}, \{-(2i-1), (2i+1)\}, \ldots, \{n+1, n-1\}, \{n+2, n\}.$$

Under the subgroup R, F generates n one-factors decomposing $K_0 \cup \Gamma(1)$.

For odd $d \neq n$, $1 \leq d \leq 2n-1$, since $gcd(d, n) = 1$, the mapping $f_d : x \mapsto dx$ is an isomorphism from $K_0 \cup \Gamma(1)$ to $K_0 \cup \Gamma(d)$. Denote $F_d = f_d(F)$. Then F_d is a one-factor and generates a one-factorization of $K_0 \cup \Gamma(d)$ under R.

For $d = n$, we construct a one-factor F_n of $K_0 \cup \Gamma(n)$:

$$\{0, n\}, \{-1, 1\}, \ldots, \{-i, i\}, \ldots, \{n+1, n-1\}.$$

Under R, F_n generates n one-factors decomposing $K_0 \cup \Gamma(n)$. □

Lemma 2.4. *Let n be a prime with $n > 3$ and u be a positive integer with $u < n$. Suppose $Q_1, \ldots, Q_u$ are u sets of pairs such that each Q_k contains pairs with different parity and belonging to different $\Gamma(d)$s. Suppose that an $SQS(2n)$ on Z_{2n} is generated by a set of blocks $\mathcal{B}$ under R such that $\mathcal{B}$ can be partitioned into u parts $P_1, \ldots, P_u$ with the property that each $P_k \cup Q_k$ is a partition of the set Z_{2n}. If there is a certain odd d_0 such that $(\cup_{1 \leq k \leq u} Q_k) \cap \Gamma(d_0) = \emptyset$, then there exists a resolvable $H(2n, 2, 4, 3)$. Moreover, there exists an $RSQS(4n)$.*

Proof. We shall show that the block set $\mathcal{B}_1 \cup \mathcal{B}_3$ of the above-mentioned $H(2n, 2, 4, 3)$ can be partitioned into parallel classes.

By Lemma 2.3, $K_0 \cup \Gamma(d_0)$ has a one-factorization. Denote its one-factors by $F_2, F_4, \ldots, F_{2n-2}, F_{d_0}$. For $d \neq d_0$, denote $\Gamma(d) = F_d$. Thus, $\{F_1, F_2, \ldots, F_{2n-1}\}$ is a one-factorization of K_{2n} on Z_{2n} and $\mathcal{B}_3 = \cup_{1 \leq j \leq 2n-1} \cup_{1 \leq k \leq n-1} \mathcal{B}(j, k)$ as before.

We distinguish $\mathcal{B}(d, k)$ if $\Gamma(d) \cap Q_k \neq \emptyset$. We may order the edges in $\Gamma(d)$ so that the $(i+k)$th edge is $\{x + 2k, y + 2k\}$ if the ith edge is $\{x, y\}$. Denote

$$\mathcal{P}_k = \{B \times \{0\}, (B + 2k) \times \{1\} : B \in P_k\} \cup \mathcal{T}_k,$$

where

$$\mathcal{T}_k = \{(E \times \{0\}) \cup ((E + 2k) \times \{1\}) : E \in Q_k\}.$$

Since $P_k \cup Q_k$ is a partition of Z_{2n}, $\mathcal{P}_k$ is a parallel class. Under R, $\mathcal{P}_k$ generates n parallel classes. For $1 \leq k \leq u$, all these nu parallel classes partition the union of $\mathcal{B}_1$

and all distinguished $\mathcal{B}(d,k)$'s. Together with the undistinguished $\mathcal{B}(d,k)$'s we obtain a resolution of $\mathcal{B}_1 \cup \mathcal{B}_3$. Thus, the $H(2n,2,4,3)$ is resolvable.

Since $\mathcal{B}_1 \cup \mathcal{B}_2 = (\mathcal{B}_1 \cup \mathcal{B}_3) \cup \mathcal{D}$ and $\mathcal{D} = \cup_{1 \le j \le 2n-1} \mathcal{B}(j,0)$, $\mathcal{B}_1 \cup \mathcal{B}_2$ is the block set of an RSQS($4n$). □

Lemma 2.5. *There exists a resolvable $H(2n,2,4,3)$ for $n \in \{5,7,13\}$.*

Proof. Apply Lemma 2.4 with n, d_0, u, P_k, and Q_k as follows.

For $n = 5$ and $d_0 = 7, u = 3$,

P_1 :	0 1 2 6	3 4 5 9	$Q_1 : 7\ 8$
P_2 :	0 2 4 7	1 3 5 8	$Q_2 : 6\ 9$
P_3 :	0 1 3 4	5 6 8 9	$Q_3 : 2\ 7$

For $n = 7$ and $d_0 = 7, u = 5$,

P_1 :	3 4 10 11	1 6 9 12	2 7 8 13	$Q_1 : 0\ 5$	
P_2 :	2 5 9 12	1 6 10 11	0 3 4 13	$Q_2 : 7\ 8$	
P_3 :	1 6 8 13	9 10 11 12	2 4 5 7	$Q_3 : 0\ 3$	
P_4 :	0 2 4 8	3 5 7 13	$Q_4 : 1\ 6$	9 12	10 11
P_5 :	6 7 10 12	0 1 9 11	$Q_5 : 2\ 3$	4 13	5 8

For $n = 13$, $d_0 = 13$ and $u = 12$ we list the first six P_k and Q_k and the other six are obtained by $P_{k+6} = \{B+1 : B \in P_k\}$ and $Q_{k+6} = \{E+1 : E \in Q_k\}$.

P_1 :	0 1 2 21	3 4 8 14	5 6 13 19	9 12 16 23	
P_2 :	0 1 16 24	9 11 14 23	2 4 10 15	3 7 17 22	
P_3 :	0 1 18 22	4 6 10 13	23 25 7 16	2 5 15 21	
P_4 :	0 1 4 12	2 3 9 17	10 11 23 7	14 16 18 5	
P_5 :	0 2 7 12	3 5 14 25	6 9 15 21	4 8 17 22	
P_6 :	0 2 20 23	22 25 6 17	4 5 7 21	8 9 14 18	10 11 19 3

Q_1 :	7 22	11 18	15 10	17 20	25 24
Q_2 :	5 8	13 6	19 18	21 12	25 20
Q_3 :	3 12	9 20	11 8	17 24	19 14
Q_4 :	13 8	15 24	19 20	21 6	25 22
Q_5 :	1 10	11 16	13 24	19 18	23 20
Q_6 :	1 12	13 16	15 24		

□

Lemma 2.6. *If there is an RSQS(v), then there exists an RSQS(pv) for $p \in \{5,7,13\}$.*

Proof. Apply Lemma 2.2 with $g = 2, u = 2p$ and $q = v/4$. Since a resolvable $H(2p,2,4,3)$ exists from Lemma 2.5, we obtain an RSQS(pv) from an RSQS(v). □

3. Constructions using resolvable candelabra quadruple systems. In this section, we give recursive constructions for RSQSs by using resolvable candelabra quadruple systems (CQSs).

CQSs are useful in the construction of SQS(v)s; see, for example, [8]. A CQS of order v with a candelabra of type $(g_1^{a_1} \cdots g_k^{a_k} : s)$, denoted by CQS$(g_1^{a_1} \cdots g_k^{a_k} : s)$, is a quadruple $(X, S, \mathcal{G}, \mathcal{A})$, where X is a set of $v = s + \sum_{1 \le i \le k} a_i g_i$ points, S is a subset of X of size s, and $\mathcal{G} = \{G_1, G_2, \ldots\}$ is a partition of $X \setminus S$ of type $g_1^{a_1} \cdots g_k^{a_k}$. The set $\mathcal{A}$ contains 4-subsets of X, called blocks, such that every 3-subset $T \subset X$ with $|T \cap (S \cup G_i)| < 3$ for all i is contained in a unique block and no 3-subset of $S \cup G_i$ is contained in any block. The members of $\mathcal{G}$ are called branches or groups, and S is called the stem of the candelabra. A CQS will be called uniform if all groups have the same size.

Now, we introduce the notion of resolvability for a CQS. A CQS$(g^n : s)$ $(X, S, \mathcal{G}, \mathcal{A})$ is said to be *resolvable* if its block set can be partitioned into $(ng(g+2s-3) + n(n-1)g^2)/6$ parts with the following two properties: (1) for each group $G \in \mathcal{G}$, there are exactly $g(g+2s-3)/6$ parts, each being a partition of $X \setminus (G \cup S)$ (called *a partial* *parallel class*); (2) there are $n(n-1)g^2/6$ parts, each being a partition of X (called *a parallel class*).

We have an easy example for resolvable CQSs. Let $(X \cup \{\infty_1, \infty_2\}, \mathcal{B})$ be an RSQS$(2n+2)$ with a resolution $P(1) \mid \cdots \mid P(n(2n+1)/3)$. There are n blocks in $\mathcal{B}$ each containing both ∞_1 and ∞_2. Denote them by $B_1, \ldots, B_n$. Without loss of generality, we assume that $B_i \in P(i)$ for $1 \le i \le n$. Denote $P'(i) = P(i) \setminus \{B_i\}$. Let $S = \{\infty_1, \infty_2\}$, $\mathcal{G} = \{B_i \backslash S : 1 \le i \le n\}$ and $\mathcal{A} = \mathcal{B} \backslash \{B_1, \ldots, B_n\}$. Take $\mathcal{G}$ as the set of groups and S as a stem. Then $(X, S, \mathcal{G}, \mathcal{A})$ is a resolvable CQS$(2^n : 2)$ with a resolution $P'(1) \mid \cdots \mid P'(n) \mid P(n+1) \mid \cdots \mid P(n(2n+1)/3)$. We state this in the following.

Lemma 3.1. *If there exists an RSQS*$(2n+2)$*, then there exists a resolvable CQS*$(2^n : 2)$.

The proofs of Hartman's theorems [7, Theorem 4.5], [6, Theorem 2.1] imply, respectively, the following two lemmas.

Lemma 3.2. *If there exists an RSQS*$(g+s : s)$*, where* $g \equiv s \pmod{12}$*, then there exists a resolvable CQS*$(g^3 : s)$.

Lemma 3.3. *If there exists an RSQS*$(g+s : s)$*, where* $g \equiv 0 \pmod{12}$ *and* $9s \ge 5g$*, then there exists a resolvable CQS*$(g^3 : s)$.

From a resolvable CQS, we can obtain an RSQS.

Lemma 3.4 (filling in holes). *Suppose that there exists a resolvable CQS*$(g^n : s)$*. If there is an RSQS*$(g+s : s)$*, then there exists an RSQS*$(ng+s : g+s)$.

Proof. Let $(X, S, \mathcal{G}, \mathcal{B})$ be the given resolvable CQS$(g^n : s)$, where $\mathcal{G} = \{G_1, \ldots, G_n\}$. Then the block set $\mathcal{B}$ has a partition $\{P(k,j) : 1 \le k \le n, 1 \le j \le g(g+2s-3)/6\} \cup \{P'(m) : 1 \le m \le n(n-1)g^2/6\}$ such that (1) for $1 \le k \le n$ and $1 \le j \le g(g+2s-3)/6$, $P(k,j)$ is a partition of $X \setminus (G_k \cup S)$; (2) for $1 \le m \le n(n-1)g^2/6$, $P'(m)$ is a parallel class on X.

For $1 \le k \le n$, construct an RSQS$(g+s : s)$ on $G_k \cup S$. Such a design exists by assumption. Denote the set of blocks in the subdesign by $\mathcal{C}$ and the set of the other blocks by $\mathcal{A}_k$. Then there are $(g+s-1)(g+s-2)/6$ parallel classes $Q(k,j)$, $1 \le j \le (g+s-1)(g+s-2)/6$, with the property for $g(g+2s-3)/6 < j \le (g+s-1)(g+s-2)/6$ each parallel class $Q(k,j)$ on $G_k \cup S$ contains a parallel class $Q'(j)$ on S.

Then $(X, \mathcal{B} \cup \mathcal{C} \cup (\cup_{1 \le k \le n} \mathcal{A}_k))$ is an SQS$(gn+s)$ and $(G_1 \cup S, \mathcal{C} \cup \mathcal{A}_1)$ is a subdesign RSQS$(g+s)$. We shall resolve this quadruple system.

The first $(s-1)(s-2)/6$ parallel classes are $P''(j) = (\cup_{1 \le k \le n}(Q(k,j) \setminus Q'(j))) \cup Q'(j)$, $g(g+2s-3)/6 < j \le (g+s-1)(g+s-2)/6$. Another $ng(g+2s-3)/6$ parallel classes are $P'(k,j) = P(k,j) \cup Q(k,j)$, $1 \le k \le n$ and $1 \le j \le g(g+2s-3)/6$. The other $n(n-1)g^2/6$ parallel classes are $P'(m)$, $1 \le m \le n(n-1)g^2/6$. Clearly, these parallel classes are pairwise disjoint. Further, the number of these parallel classes is $(s-1)(s-2)/6 + ng(g+2s-3)/6 + n(n-1)g^2/6$, which is the required number of parallel classes in an RSQS$(gn+s)$. Therefore, such an SQS is also resolvable. Further, since $Q(1,j) \subset P''(j)$ for $g(g+2s-3)/6 < j \le (g+s-1)(g+s-2)/6$ and $Q(1,j) \subset P'(1,j)$ for $1 \le j \le g(g+2s-3)/6$, so this RSQS$(ng+s)$ is also an RSQS$(ng+s : g+s)$. □

This lemma shows that resolvable CQS is useful for the construction of RSQS. We give other constructions for resolvable CQS as follows.

Lemma 3.5. *Suppose that there exists an RSQS*$(u+1)$*. If there is a resolvable CQS*$(g^3 : s)$*, then there is a resolvable CQS*$(g^u : s)$.

Proof. Let $(X \cup \{\infty\}, \mathcal{B})$ be the given RSQS(u) with a resolution $P_1 \mid \cdots \mid P_{u(u-1)/6}$. We shall construct the desired design on $Y = (X \times Z_g) \cup S$, where $S \cap (X \times Z_g) = \emptyset$ and $|S| = s$.

For each block $B \in \mathcal{B}$ with $\infty \notin B$, construct a resolvable $H(4, g, 4, 3)$ on $B \times Z_g$ with groups $\{x\} \times Z_g$, $x \in B$. Such a design exists by Lemma 2.1. Denote its block set by $\mathcal{C}_B$ and its parallel classes by $P'_B(m)$, $1 \leq m \leq g^2$.

For each block $B \in \mathcal{B}$ with $\infty \in B$, construct a resolvable CQS$(g^3 : s)$ on $((B \setminus \{\infty\}) \times Z_g) \cup S$. Such a design exists by assumption. Denote its block set by $\mathcal{A}_B$. Then there is a resolution $\{P_B(x, j) : x \in B \setminus \{\infty\}, 1 \leq j \leq g(g+2s-3)/6\} \cup \{P''_B(m) : 1 \leq m \leq g^2\}$ such that (1) for $x \in B \setminus \{\infty\}$ and $1 \leq j \leq g(g + 2s - 3)/6$, $P_B(x, j)$ is a partition of $(B \setminus \{x, \infty\}) \times Z_g$; (2) for $1 \leq m \leq g^2$, $P''_B(m)$ is a parallel class on $((B \setminus \{\infty\}) \times Z_g) \cup S$.

By [8, Theorem 1.4], $(\cup_{B \in \mathcal{B}, \infty \notin B} \mathcal{C}_B) \cup (\cup_{B \in \mathcal{B}, \infty \in B} \mathcal{A}_B)$ is the block set of a CQS$(g^u : s)$. It remains to give its resolution with the two properties.

For $x \in X$ and $1 \leq j \leq g(g + 2s - 3)/6$, let $P(x, j) = \cup_{B \in \mathcal{B}, \{x, \infty\} \subset B} P_B(x, j)$. Then each $P(x, j)$ is a partition of $(X \setminus \{x\}) \times Z_g$. For $1 \leq k \leq u(u - 1)/6$ and $1 \leq m \leq g^2$, let $P'(k, m) = (\cup_{B \in P_k,\ \infty \in B} P''_B(m)) \cup (\cup_{B \in P_k,\ \infty \notin B} P'_B(m))$. Then each $P'(k, m)$ is a parallel class on Y. It has been checked that all these parallel classes form the required resolution. It follows that a resolvable CQS$(g^u : s)$ exists. □

Before we give a tripling construction for resolvable CQS, we first give the construction of a resolvable CQS$(6^3 : 2)$.

Lemma 3.6. *There exists a resolvable CQS*$(6^3 : 2)$.

Proof. We start with a CQS$(3^3 : 1)$ on $Z_9 \cup \{\infty\}$ with groups $G_i = \{i, i+3, i+6\}$, $0 \leq i \leq 2$, and a stem $S = \{\infty\}$, whose block set $\mathcal{B}$ is generated by the following 9 base blocks under the mapping $x \rightarrow x + 3 \pmod 9$. (Note that such a CQS exists by [3, Theorem 5.1].)

$\mathcal{A}_\infty$:	$0\ 1\ 2\ \infty$	$0\ 4\ 8\ \infty$	$0\ 5\ 7\ \infty$
$\mathcal{A}_1$:	$1\ 3\ 2\ 6$	$1\ 3\ 5\ 7$	$2\ 6\ 5\ 7$
$\mathcal{A}_2$:	$4\ 7\ 5\ 8$	$3\ 6\ 5\ 8$	$3\ 6\ 4\ 7$

We consider each base block as an ordered quadruple given above so that each block $B \in \mathcal{B}$ is then ordered.

The desired design will be based on $X = (Z_9 \cup \{\infty\}) \times Z_2$ with groups $G'_i = G_i \times Z_2$, $0 \leq i \leq 2$, and a stem $S' = S \times Z_2$.

For each block $B = \{x_1, x_2, x_3, \infty\} \in \mathcal{B}$, construct a resolvable CQS$(2^3 : 2)$ on $B \times Z_2$ with groups $\{x_i\} \times Z_2$, $1 \leq i \leq 3$, and a stem S'. Such a design exists by Lemma 3.1. Denote its block set by $\mathcal{A}_B$. Then $\mathcal{A}_B$ has a partition $\{P_B(x_1), P_B(x_2), P_B(x_3)\} \cup \{P_B(r', r) : r', r \in Z_2\}$ such that (1) for $1 \leq k \leq 3$, $P_B(x_k)$ contains one block $(\{x_1, x_2, x_3\} \setminus \{x_k\}) \times Z_2$; (2) each $P_B(r', r)$ is a parallel class $B \times Z_2$.

For each block $B = \{a', b', c', d'\} \in \mathcal{B}$ with its four elements in the given order and $\infty \notin B$ we shall construct a special $H(4, 2, 4, 3)$ on $B \times Z_2$ with groups $\{x\} \times Z_2$, $x \in B$. Denote

$$C_B(k, i, j) = \{(a', i), (b', i + k), (c', j), (d', j + k)\},$$

$$\mathcal{C}_B(k) = \{C_B(k, i, j) : i, j \in Z_2\}.$$

Then $\mathcal{C}_B = \mathcal{C}_B(0) \cup \mathcal{C}_B(1)$ is the block set of the $H(4, 2, 4, 3)$.

Let $\mathcal{D} = (\cup_{B \in \mathcal{B}, \infty \notin B} \mathcal{C}_B) \cup (\cup_{B \in \mathcal{B}, \infty \in B} \mathcal{A}_B)$. Then by [10, Theorem 7.1] $(X, S', \{G'_i : 0 \leq i \leq 2\}, \mathcal{D})$ is a CQS$(6^3 : 2)$. It remains to show that it is also resolvable. It should contain 36 parallel classes on X and 7 partial parallel classes on $X \setminus (G'_i \cup S')$ for any $0 \leq i \leq 2$.

For $0 \le i \le 2$ and $x \in G_i$, let $P(x) = \cup_{B\in\mathcal{B},\{\infty,x\}\subset B} P_B(x)$. Then $P(x)$ is a partial parallel class on $X \setminus (G_i' \cup S')$. For $r', r \in Z_2$, the other four partial parallel classes $P'(i, r', r)$ on $X \setminus (G_i' \cup S')$ are obtained as follows.

Denote the three base blocks of $\mathcal{A}_2$ by B_0, B_1, B_2 in order. For $0 \le i \le 2$, let $\mathcal{B}_i = \{3j + B_i : 0 \le j \le 2\}$ and for $r', r \in Z_2$ let

$$P'(i, r', r) = \{C_B(1, r', r) : B \in \mathcal{B}_i\}.$$

Then $P'(i, r', r)$ is a partial parallel class on $X \setminus (G_i' \cup S')$. Note that for $0 \le i \le 2$, $\cup_{r',r\in Z_2} P'(i, r', r) = \cup_{B\in\mathcal{B}_i} \mathcal{C}_B(1)$.

Now, we give the required 36 parallel classes $P''(i, j, r', r)$ on X, where $0 \le i, j \le 2$ and $r', r \in Z_2$. Denote the three base blocks of $\mathcal{A}_1$ by A_0, A_1, A_2 in order. Let $D_0 = A_0, D_1 = A_1 + 3 = 4681, D_2 = A_2 + 6 = 8324$. Let $\mathcal{A}(i, 0)$ be as follows and $\mathcal{A}(i, j) = \{3j + B : B \in \mathcal{A}(i, 0)\}$ for $0 \le j \le 2$.

$$\begin{aligned}
\mathcal{A}(1,0) &= \{0\ 4\ 8\ \infty, \quad A_0, \quad A_1, \quad A_2\},\\
\mathcal{A}(2,0) &= \{0\ 1\ 2\ \infty, \quad B_0, \quad B_1, \quad B_2\},\\
\mathcal{A}(0,0) &= \{0\ 5\ 7\ \infty, \quad D_0, \quad D_1, \quad D_2\}.
\end{aligned}$$

Let

$$P(1, j, r', r) = \{C_{A_0+3j}(0, r', r' + r), C_{A_1+3j}(0, r' + 1, r), C_{A_2+3j}(0, r' + r + 1, r + 1)\},$$

$$P(2, j, r', r) = \{C_{B_0+3j}(0, r' + r, r'), C_{B_1+3j}(0, r, r' + 1), C_{B_2+3j}(0, r + 1, r' + r + 1)\},$$

$$P(0, j, r', r) = \{C_{D_0+3j}(1, r', r'+r), C_{D_1+3j}(1, r'+r+1, r'), C_{D_2+3j}(1, r'+1, r'+r+1)\}.$$

Let $P''(i, j, r', r) = P_B(r', r) \cup P(i, j, r', r)$, where $B \in \mathcal{A}(i, j)$ and $\infty \in B$. Since in each $\mathcal{A}(i, j)$, each element occurs twice if it is not in the block containing ∞, each $P''(i, j, r', r)$ is a parallel class on X. For $0 \le i, j \le 2$, let $\mathcal{A}(i, j)' = \{B \in \mathcal{A}(i, j) : \infty \notin B\}$. Note that for $0 \le j \le 2$, $\cup_{r',r\in Z_2} P(i, j, r', r) = \cup_{B\in\mathcal{A}(i,j)'} \mathcal{C}_B(0)$ for $i \in \{1, 2\}$ and $\cup_{r',r\in Z_2} P(0, j, r', r) = \cup_{B\in\mathcal{A}(0,j)'} \mathcal{C}_B(1)$. Since $\cup_{0\le j\le 2}(\mathcal{A}(0, j)' \cup \mathcal{B}_j) = \{B \in \mathcal{B} : \infty \notin B\}$, we have

$$\bigcup_{B\in\mathcal{B},\infty\notin B} \mathcal{C}_B(1) = \bigcup_{0\le j\le 2} \ \bigcup_{r',r\in Z_2} (P'(j, r', r) \cup P(0, j, r', r)).$$

We have proved that $\mathcal{D}$ has the resolution $\{P(x) : x \in Z_9\} \cup \{P'(i, r', r) : 0 \le i \le 2, r', r \in Z_2\} \cup \{P''(i, j, r', r) : 0 \le i, j \le 2, r', r \in Z_2\}$, so the CQS$(6^3 : 2)$ is resolvable. □

Lemma 3.7. *If there is a resolvable CQS*$(g^3 : s)$ *for* $g \equiv s \equiv 0 \pmod 2)$, *then there exists a resolvable CQS*$((3g)^3 : s)$.

Proof. We keep the notation of Lemma 3.6 and we adapt the proof to the present situation. Let $g = 2m$ and $s = 2n$. The desired design will be based on $Y = (Z_9 \times Z_2 \times Z_m) \cup (\{\infty\} \times Z_2 \times Z_n)$ with groups $G_i'' = G_i' \times Z_m$, $0 \le i \le 2$, and a stem $S'' = S' \times Z_n$.

For each block $B = \{x_1, x_2, x_3, \infty\} \in \mathcal{B}$, construct a resolvable CQS$(g^3 : s)$ on $(\{x_1, x_2, x_3\} \times Z_2 \times Z_m) \cup S''$ with groups $\{x_i\} \times Z_2 \times Z_m$, $1 \le i \le 3$ and a stem S''. Such a design exists by assumption. Denote its block set by $\mathcal{A}_B''$. Then $\mathcal{A}_B''$ has a partition $\{P_B(x_i, \ell) : 1 \le i \le 3,\ 1 \le \ell \le g(g+2s-3)/6\} \cup \{P_B(r', r, h) : 1 \le h \le m^2, r', r \in Z_2\}$ such that (1) for any k and ℓ, $P_B(x_k, \ell)$ is a partition of $(\{x_1, x_2, x_3\} \setminus \{x_k\}) \times Z_2 \times Z_m$; (2) each $P_B(r', r, h)$ is a parallel class on $((B \setminus \{\infty\}) \times Z_2 \times Z_m) \cup S''$.

For each block $B \in \mathcal{B}$ and $\infty \notin B$, we shall construct a special $H(4, g, 4, 3)$ on $B \times Z_2 \times Z_m$ with groups $\{x\} \times Z_2 \times Z_m$, $x \in B$. Denote its block set by $\mathcal{C}_B''$, which can be obtained as follows. Note that $\mathcal{C}_B$ is the block set of the special $H(4,2,4,3)$ on $B \times Z_2$. For any $A \in \mathcal{C}_B$, from Lemma 2.1 we have a resolvable $H(4,m,4,3)$ on $A \times Z_m$ with groups $\{a\} \times Z_m, a \in A$. Denote its block set by $\mathcal{B}(A)$ and the m^2 parallel classes by $P(A,h), 1 \le h \le m^2$. Then we have

$$\mathcal{C}''_B = \bigcup_{A \in \mathcal{C}_B} \mathcal{B}(A).$$

Let $\mathcal{D}'' = (\cup_{B\in\mathcal{B},\infty\notin B}\mathcal{C}''_B) \cup (\cup_{B\in\mathcal{B},\infty\in B}\mathcal{A}''_B)$. Then by [10, Theorem 7.1] $(Y, S''$, $\{G''_i : 0 \le i \le 2\}, \mathcal{D}'')$ is a CQS$((3g)^3 : s)$. It remains to show that it is also resolvable. It should contain $9g^2$ parallel classes on Y and $g(3g+2s-3)/2$ partial parallel classes on $Y \setminus (G''_i \cup S'')$ for any i, $0 \le i \le 2$.

For $1 \le \ell \le g(g+2s-3)/6$, $0 \le i \le 2$, $x \in G_i$, let $P(x,\ell) = \cup_{B\in\mathcal{B},\{x,\infty\}\subset B} P(x,\ell)$. Then $P(x,\ell)$ is a partition of $Y \setminus (G''_i \cup S'')$. The other g^2 partial classes $P'(i,r',r,h)$ on $Y \setminus (G''_i \cup S'')$ can be obtained from $P'(i,r',r)$, where

$$P'(i,r',r,h) = \bigcup_{A\in P'(i,r',r)} P(A,h)$$

and $r', r \in Z_2, 1 \le h \le m^2$.

Now, we give the required $9g^2$ parallel classes $P''(i,j,r',r,h)$ on Y. For $0 \le i,j \le 2$, $r', r \in Z_2$, and $1 \le h \le m^2$, denote $P(i,j,r',r,h) = \cup_{A\in P(i,j,r',r)} P(A,h)$ and let

$$P''(i,j,r',r,h) = P_B(r',r,h) \cup P(i,j,r',r,h),$$

where $B \in \mathcal{A}(i,j)$ and $\infty \in B$. Since $P_B(r',r,h)$ is a partition of $((B \setminus \{\infty\}) \times Z_2 \times Z_m) \cup S''$ and $P(i,j,r',r,h)$ is a partition of $(Z_9 \setminus B) \times Z_2 \times Z_m$, then $P''(i,j,r',r,h)$ is a parallel class on Y.

Since $\mathcal{C}''_B = \cup_{A\in\mathcal{C}_B}\mathcal{B}(A)$, $\mathcal{D}''$ has the resolution $\{P(x,\ell) : x \in Z_9, 1 \le \ell \le g(g+2s-3)/6\} \cup \{P'(i,r',r,h) : 0 \le i \le 2, r', r \in Z_2, 1 \le h \le m^2\} \cup \{P''(i,j,r',r,h) : 0 \le i,j \le 2, r', r \in Z_2, 1 \le h \le m^2\}$, and the CQS$((3g)^3 : s)$ is resolvable. □

Lemma 3.8. *If there is an RSQS*(v)*, then there exists an RSQS*$(6v-4)$.

Proof. We start with an RSQS(v). We apply Lemma 3.5 with the known resolvable CQS$(6^3 : 2)$ in Lemma 3.6. Then we obtain a resolvable CQS$(6^{v-1} : 2)$. Further, apply Lemma 3.4 with the known RSQS$(8 : 2)$. Then we obtain an RSQS $(6v-4)$. □

4. Small designs. In this section, we construct the desired 23 RSQSs.

Lemma 4.1. *There exists an RSQS*(v) *for* $v \in \{236, 596, 1100, 1820, 2324, 2540,$ $3620,\ 6356\}$.

Proof. Since there exists an RSQS$((v+4)/6)$ by Theorem 1.3, the conclusion then follows from Lemma 3.8. □

Lemma 4.2. *There is an RSQS*(v) *for* $v \in \{220, 364, 460, 676, 1316, 2236, 2380,$ $2740, 3820\}$.

Proof. For $v \in \{220, 460, 2380, 2740, 3820\}$, the result follows from Lemma 2.6 with $p = 5$ and the known RSQS(v/p) in Theorem 1.3.

For $v \in \{364, 1316\}$, the result follows from Lemma 2.6 with $p = 7$ and the known RSQS(v/p) in Theorem 1.3.

For $v \in \{676, 2236\}$, the result follows from Lemma 2.6 with $p = 13$ and the known RSQS(v/p) in Theorem 1.3. □

Lemma 4.3. *There exists an RSQS*(v) *for* $v \in \{724, 1076, 1252, 2308, 2812\}$.

Proof. For $v = 724$ we start with an RSQS(20 : 4), which exists by Theorem 1.3. By Lemma 3.2 a resolvable CQS(16^3 : 4) exists. Then by Lemma 3.7 there is a resolvable CQS(48^3 : 4), which together with an RSQS(16) and an application of Lemma 3.5 (take $g = 48$, $s = 4$ and $u = 15$) gives a resolvable CQS(48^{15} : 4). Fill the holes of this CQS with an RSQS(52 : 4) to get an RSQS(724).

For $v = 1076$ we start with an RSQS(20 : 8) in [7]. By Theorem 1.2 there is an RSQS(44 : 8). Since there is an RSQS(44 : 4), by Lemma 3.2 there is a resolvable CQS(40^3 : 4). Applying Lemma 3.4 gives an RSQS(124 : 44), which together with an RSQS(44 : 8) leads to an RSQS(124 : 8). From the above resolvable CQS(40^3 : 4) we can also obtain a resolvable CQS(120^3 : 4) by Lemma 3.7. Further, applying Lemma 3.4 with the known RSQS(124 : 4) in Theorem 1.3 yields an RSQS(364 : 124), which together with the above RSQS(124 : 8) gives an RSQS(364 : 8). Then we apply Theorem 1.1 to give an RSQS(1076).

For $v = 1252$ we start with an RSQS(8), which exists by Theorem 1.3. Applying the doubling construction yields an RSQS(16 : 8). Since there is an RSQS(52 : 16) in [6], there is an RSQS(52 : 8). From the above resolvable CQS(48^3 : 4) we may apply Lemma 3.4 to give an RSQS(148 : 52). So, there are an RSQS(148 : 16) and an RSQS(148 : 8). By Lemma 3.2 there is an CQS(140^3 : 8). Further, applying Lemma 3.4 gives an RSQS(428 : 148), which together with an RSQS(148 : 16) leads to an RSQS(428 : 16). It follows from Lemma 3.2 that a resolvable CQS(412^3 : 16) exists. An RSQS(1252) is then obtained by Lemma 3.4.

For $v = 2308$, since there is an RSQS(260 : 4), by Lemma 3.2 there is a resolvable CQS(256^3 : 4). Applying Lemma 3.7 gives a resolvable CQS(768^3 : 4). We then apply Lemma 3.4 with the known RSQS(772 : 4) to obtain an RSQS(2308).

For $v = 2812$, by Lemma 2.5 there is a resolvable $H(10, 2, 4, 3)$. Applying Lemma 2.2 with $q = 19$, we obtain an RSQS(380). From the proof of Lemma 2.2 such an RSQS(380) contains a subdesign RSQS(76), i.e., an RSQS(380 : 76) exists. By Lemma 3.2 we have a resolvable CQS(304^3 : 76). Applying Lemma 3.4 gives an RSQS(988 : 76). From the last CQS we apply Lemma 3.7 to obtain a resolvable CQS(912^3 : 76). Then, an RSQS(2812) exists by Lemma 3.4. □

LEMMA 4.4. *There exists a resolvable* $H(146, 2, 4, 3)$ *and an RSQS*(292).

Proof. Apply Lemma 2.4 with $n = 73, d_0 = 73$, and $u = 56$. We list three pairs of (P_k, Q_k) below. For $k = 2$, the pair generates 9 pairs under the mapping $f_i : x \mapsto 5^i x$ for $0 \le i \le 8$. For $k = 3$, the pair generates 18 pairs under the mapping $f_i : x \mapsto 5^i x$ for $0 \le i \le 17$. We then obtain 28 pairs of (P_k, Q_k). Under the mapping $\delta : \Delta \mapsto \Delta + 1$, they lead to the other 28 pairs.

P_1 contains the following 24 base blocks:

0 2 18 104	1 11 91 83	4 54 16 122	3 107 63 9	6 88 14 36
20 26 74 40	5 35 129 105	8 12 44 70	7 27 41 25	10 110 34 100
30 48 46 90	21 111 101 29	66 78 28 106	15 75 117 69	60 68 132 38
64 118 112 98	89 67 37 113	50 86 82 24	45 79 59 61	56 80 126 136
13 131 53 17	19 81 139 31	57 97 125 93	77 51 135 39	

P_2 contains the following 30 base blocks:

0 1 61 62	108 120 122 134	2 85 101 38	138 140 24 26	3 30 44 71
8 91 28 18	136 110 130 104	14 16 20 22	94 114 82 102	12 32 72 92
74 76 84 86	96 116 124 144	80 100 34 54	17 119 89 45	40 42 68 70
11 65 27 81	25 127 139 95	7 58 52 103	141 49 129 37	19 73 143 51
4 5 6 78	135 21 75 107	9 36 63 109	93 121 77 105	98 99 125 126
13 64 115 137	15 117 29 131	33 87 59 113	97 53 123 79	118 55 23 106

P_3 contains the following 32 base blocks:

0 1 3 4	2 29 83 110	6 7 11 12	8 35 143 24	88 89 129 130
16 17 25 26	18 45 115 142	30 31 41 42	28 55 33 60	82 109 107 134
48 49 63 64	46 73 13 40	58 59 75 76	44 71 65 92	98 99 117 118
10 37 53 80	78 105 23 50	90 91 121 122	66 93 5 32	102 103 139 140
14 15 21 22	108 135 69 96	84 85 131 132	54 81 9 36	86 87 137 138
38 39 51 52	56 57 111 112	94 95 119 120	20 47 79 106	114 141 43 70
100 127 77 104	34 61 97 124			

Q_1 contains the following 25 pairs:

22 23	32 43	33 42	47 52	49 62	55 58	65 72	71 92
76 119	84 123	87 134	94 143	95 128	96 127	99 140	103 138
109 124	85 130	114 133	115 142	116 141	120 137	121 144	73 102
108 145							

Q_2 contains the following 13 pairs:

10 31	35 46	39 48	41 56	43 50	47 60	57 88	66 67	69 132
83 112	90 133	111 128	142 145					

Q_3 contains the following 9 pairs:

19 62	27 68	67 72	74 101	113 116	123 136	125 126	128 145
133 144 □							

The 23 undecided orders in Theorem 1.3 have been solved in Lemmas 4.1–4.4. By Theorem 1.3, we have proved Theorem 1.4.

Acknowledgment. We would like to thank the referees for many helpful comments.

REFERENCES

[1] T. R. Booth, *A resolvable quadruple system of order* 20, Ars Combin., 5 (1978), pp. 121–125.

[2] D. L. Greenwell and C. C. Lindner, *Some remarks on resolvable quadruple systems*, Ars Combin., 6 (1978), pp. 215–221.

[3] H. Hanani, *A class of three-designs*, J. Combin. Theory Ser. A, 26 (1979), pp. 1–19.

[4] H. Hanani, *On quadruple systems*, Canad. J. Math., 12 (1960), pp. 145–157.

[5] A. Hartman, *Resolvable Steiner quadruple systems*, Ars Combin., 9 (1980), pp. 263–273.

[6] A. Hartman, *The existence of resolvable Steiner quadruple systems*, J. Combin. Theory Ser. A, 44 (1987), pp. 182–206.

[7] A. Hartman, *Tripling quadruple systems*, Ars Combin., 10 (1980), pp. 255–309.

[8] A. Hartman and K. T. Phelps, *Steiner quadruple systems*, in Contemporary Design Theory, J. H. Dinitz and D. R. Stinson, eds., John Wiley, New York, 1992, pp. 205–240.

[9] W. H. Mills, *On the existence of H designs*, Congr. Numer., 79 (1990), pp. 129–141.

[10] H. Mohácsy and D. K. Ray-Chaudhuri, *Candelabra systems and designs*, J. Statist. Plann. Inference, 106 (2002), pp. 419–448.

SIAM J. Discrete Math.
Vol. 19, No. 2, pp. 431–447

BOUNDS ON THE TRAVEL COST OF A MARS ROVER PROTOTYPE SEARCH HEURISTIC[*]

APURVA MUDGAL[†], CRAIG TOVEY[†], SAM GREENBERG[‡], AND SVEN KOENIG[§]

Abstract. D^* is a greedy heuristic planning method that is widely used in robotics, including several Nomad class robots and the Mars rover prototype, to reach a destination in unknown terrain. We obtain nearly sharp lower and upper bounds of $\Omega(n \log n / \log \log n)$ and $O(n \log n)$, respectively, on the worst-case total distance traveled by the robot, for the grid graphs on n vertices typically used in robotics applications. For arbitrary graphs we prove an $O(n \log^2 n)$ upper bound.

Key words. robot travel, D^*, grid graph, girth, planar graph, search heuristic, Mars rover, greedy algorithm

AMS subject classifications. 68W40, 68T20

DOI. 10.1137/S089548010444256X

1. Introduction.

D^* is a greedy heuristic planning method that is widely used to direct a robot in a terrain with initially unknown obstacles from given start to given goal coordinates. D^* always moves the robot along a shortest presumed unblocked path from its current coordinates to the goal coordinates, presuming that as-yet-unobserved portions of the terrain have no obstacles. It stops when it has reached the goal coordinates or determined that this is impossible. If movement along the current path is blocked by an obstacle, the shortest presumed unblocked path changes and D^* needs to replan. This can be implemented efficiently [11] and easily [4].

In robotics applications, the continuous terrain is usually discretized into a grid. Robot movement then corresponds to traversal from vertex to adjacent vertex in a grid graph. The graph is known in the sense that the vertices (grid cells) and edges are known. Impassable features of the terrain, which determine the graph's structure, may be known via satellite reconnaissance, prior exploration, or mapping. The graph is unknown in the sense that vertices of the graph may be blocked by debris, crevices, or other obstacles. An obstacle is not known until the robot's sensors detect it, for example, as the robot attempts to move to it.

D^* is also used in other AI applications to reach a desired goal state from an initial starting state [13, 3, 7, 14]. In these applications, and in some terrains such as buildings, the graph may be a Voronoi or other type of graph rather than a grid graph. In all of these applications the vertices can be recognized—in the case of robot movement, by the physical coordinates; in other planning problems by state identifiers.

[*] Received by the editors March 29, 2004; accepted for publication (in revised form) April 15, 2005; published electronically November 4, 2005. This research was partly supported by NSF awards under contracts IIS-9984827, IIS-0098807, and ITR/AP-0113881. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations and agencies or the U.S. Government.

http://www.siam.org/journals/sidma/19-2/44256.html

[†] College of Computing, Georgia Institute of Technology, Atlanta, GA 30332-0280 (apurva@cc.gatech.edu, ctovey@cc.gatech.edu).

[‡] School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332-0160 (SamIAm@math.gatech.edu).

[§] Computer Science Department, USC, Los Angeles, CA 90089-0781 (skoenig@usc.edu).

The D^* algorithm has some advantages over depth first search (DFS) in practice, including ease of replanning if the robot is moved to a new location, empirically good average performance, and effective use of partial terrain information [6]. D^* has been used outdoors on an autonomous high-mobility multiwheeled vehicle that navigated 1,410 meters to the goal location in an unknown area of flat terrain with sparse mounds of slag as well as trees, bushes, rocks, and debris [13]. As a result of this demonstration, D^* is now widely used in the DARPA unmanned ground vehicle (UGV) program, for example, on the UGV Demo II vehicles. D^* is also being integrated into a Mars rover prototype (according to Anthony Stentz), tactical mobile robot prototypes, and other military robot prototypes for urban reconnaissance [3, 7, 14]. Furthermore, it has been used indoors on Nomad 150 mobile robots in robot-programming classes to reach a goal location in unknown mazes [9, 8]. D^* has also been used as the key method in various robot-navigation software [2, 12].

Given its simple form and many applications it would be quite interesting to know analytically how well D^* performs. The measure by which we assess performance here is the worst-case distance traveled by the robot. We focus on travel distance in the terrain rather than travel planning time because robots move so slowly that the task-completion times are completely dominated by their travel times.

For the rest of the paper, n denotes the number of vertices in the terrain graph $G = (V, E)$. In practice, D^* seems to perform reasonably well and, in many domains, exhibits a performance that is linear in n [6], i.e., the same order as DFS, but it is not known whether this is due to properties of the test terrains or whether the plan-execution times are indeed guaranteed to be good on any terrain. However, in [6] it was also shown that for arbitrary graphs the performance is $\Omega(n \log n/ \log\log n)$. Here we prove the same $\Omega(n \log n/ \log\log n)$ bound for grid graphs. The proof is a considerably modified version of the construction in [6]. This establishes that D^* has superlinear worst-case performance on the class of graphs used in real robotics applications.

The best upper bound on D^* previously known was $O(n^{3/2})$ [5]. We prove an upper bound of $O(n \log n)$ for planar graphs. This leaves only a $\log\log n$ gap, and establishes that D^* is only slightly inferior to DFS in this worst-case performance sense. As mentioned above, D^* is also employed for other applications in which the graph may not possess the grid structure. For arbitrary graphs we prove an upper bound of $O(n \log^2 n)$. Thus D^* has a rather good performance guarantee in general.

In sections 2–4 we assume that the robot has tactile (short-range) sensors. In section 5.1 we extend the results to long-range sensors. In particular, the lower bound applies to any line-of-sight sensor, and the upper bounds apply to all sensor types. In section 5.2 we extend results to the case where both vertices and edges may be blocked.

2. Definitions. We assume that the robot is equipped with a tactile (short-distance) sensor, omni-directional, point-sized, and capable of error-free motion and sensing. The sensors on board the robot uniquely identify its location. We model the terrain as a graph. Vertices in the graph represent locations in the terrain. Traversing an edge in the graph corresponds to traveling from one location to an adjacent location in the terrain. We are interested in the quality of the plans determined by D^* as a function of the number of vertices of the graph.

With these assumptions, we can formalize the behavior of D^* as follows. We call a graph $H = (V, E)$ vertex-blocked by $B \subset V$ if B is the set of blocked vertices, vertices that cannot be traversed. On a finite undirected graph $H = (V, E)$ vertex-blocked by

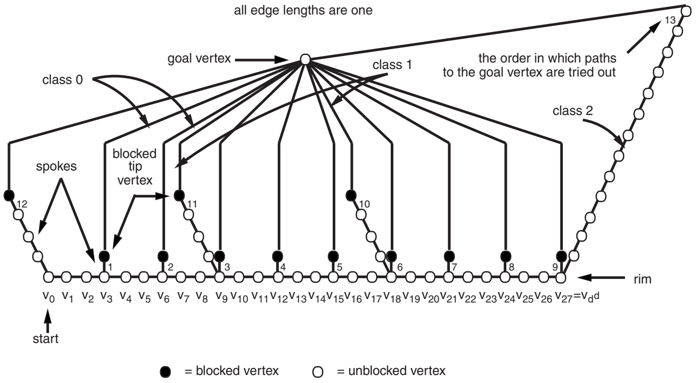


FIG. 1. *Reference* [6]*'s example graph for lower bound.*

B, a robot has to reach a designated goal vertex t from a start vertex s. D^* always moves the robot from its current vertex along a shortest presumed unblocked path to the goal vertex. A presumed unblocked path is one that contains no vertices which are known to be blocked. Initially, the robot has no information about B except that $s \notin B$. If the robot attempts to move to a blocked vertex v, it learns that $v \in B$. D^* then recomputes a new presumed unblocked path to begin the next iteration. D^* terminates when the robot reaches the goal vertex or there are no presumed unblocked paths to the goal vertex, in which case the goal vertex is unreachable from the start vertex. Additional notation to formalize the information state of the robot is given in section 4.1.

3. D^*: Lower bound on grids.

We now prove a lower bound on the worst-case travel distance of D^* on vertex-blocked grids. First, we review the construction of [10, 6], which employs the key idea of tricking the robot into traversing the same long path back and forth many times. Second, we give an overview of how to transform that example into a grid without losing the key idea. Third, we explain exactly how the grid is constructed. Last, we analyze the worst-case travel distance of D^* on our grid graph, proving the lower bound.

3.1. Making the robot go to and fro.[1]

The analysis of [10, 6] proved that the worst-case travel distance of D^* is $\Omega(\frac{n \log n}{\log \log n})$ steps on vertex-blocked graphs $H = (V, E)$. This lower bound is achieved with graphs of the structure shown in Figure 1. We now sketch the main idea of its construction, but with our own "rim-and-spoke" terminology, in order to introduce our much more complex grid construction.

The graph of Figure 1 consists of a long horizontal path of length d^d (where d is an integer parameter), which we call the "rim," and a set of "spokes" of varying lengths attached to the rim at various vertices. The uppermost "tip" vertex of each spoke is blocked and connected to the goal vertex by an edge. Note that the edges from the tips to the goal are physically unrealistic edges, because they allow the robot

[1]

A charming old bear at the zoo
Could always find something to do
  When tired, you know
  Of the walk to and fro
He'd reverse, and walk fro and to.

to move from any tip to the goal in one step. The possible spoke lengths are $\sum_{i=0}^{h} d^i$ for the nonnegative integers $h = 0 \cdots d-1$. We refer to a spoke of length $\sum_{i=0}^{h} d^i$ as a "class h spoke." Longer spokes are spaced farther apart from each other than short spokes. In particular, the vertices where class h spokes attach to the rim have distance d^{h+1} from each other. Hence, if the robot is at a vertex where a class h spoke attaches to the rim, then it is shorter to go to the goal along the rim to the next class h spoke than it is to go via any class $h+1$ spoke.

In particular, in Figure 1 there are three classes of spokes: 0, 1, and 2. The robot does not know that the shortest unblocked path to the goal from starting vertex v_0 is to traverse the rim to v_{27}, then the long class 2 spoke, and reach the goal vertex. Instead, the robot tries to reach the goal through the shortest presumed unblocked path via the short class 0 spoke at v_3, then the class 0 spoke at v_6, and so on until it tries the class 0 spoke at the right end of the rim, v_{27}. From there, the shortest presumed unblocked path to the goal is via the class 1 spoke at v_{18}. Thus the robot is led to traverse the rim from right to left, checking each class 1 spoke. Finally, the robot traverses the rim a third time, reaching the goal via the class 2 spoke.

In general, the robot starts at vertex v_0; it traverses the rim from left to right, checking the class 0 spokes for a path to the goal vertex; then it returns along the rim from right to left, checking class 1 spokes for a path to the goal vertex, and so on. Each class forces the robot to traverse the rim once. Thus the total travel distance is $\geq d^{d+1}$. A computation shows that there are $O(d^d)$ vertices in the graph, and hence the total travel distance is $\Omega(\frac{n \log n}{\log \log n})$.

3.2. Conceptual overview. We wish to construct a grid that captures the key idea from the previous analysis: to fool the robot into traversing a lengthy rim many times by visiting all the class h spokes before visiting any class $h+1$ spokes. However, the graph topology of the previous analysis cannot directly be embedded into a grid; the goal vertex must be simultaneously adjacent to the ends of many spokes of greatly different lengths, which moreover are placed at great distances from each other. In a grid, on the other hand, each cell is adjacent to at most four cells, and adjacent cells are physically close. We use several ideas to modify the graph topology of the previous construction to be able to embed it into a grid. A conceptual sketch of these ideas is shown in Figure 2.

1. Attach each spoke at a separate vertex to the rim (Figure 2a). This eliminates the problem of a vertex on the rim being adjacent to too many other vertices. As long as longer class spokes are spaced far enough apart, the robot is still fooled into repeatedly traversing the rim.
2. Remove the very short spokes (Figure 2b). We must place the goal vertex at some distance D from the rim, and we thus cannot construct spokes of length less than D. In particular, we only use classes $0.8d$ to $0.9d$ instead of using classes 0 to d.
3. Move the spokes physically closer together, but maintain their distances from each other along the rim. We do this by "squeezing" the rim into an accordion shape (Figure 2c). In particular, the sections of the rim between spokes get bent into long loops, which we call "columns."
4. Redesign the spokes so that they all have the same physical height, while maintaining their original lengths (Figure 2c). In particular, build a pair of blocked walls of the same height, with some space between them. Put a twisty path of the appropriate length in between the walls.
5. Once the spokes are fairly close and of equal height, bend the rim into part of

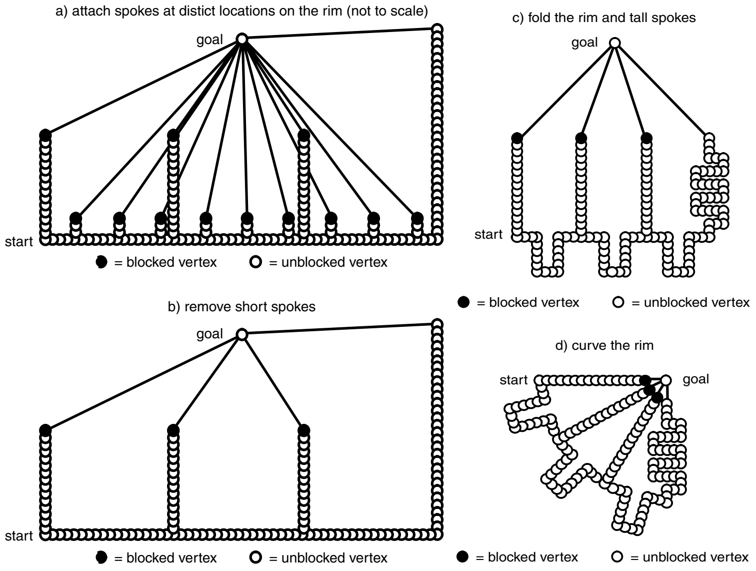


FIG. 2. *Steps of the transformation.*

a circular arc, pushing the tips of the spokes together towards the goal vertex (Figure 2d). It is not possible to squeeze too many distinct vertices into a small area on a grid, but this problem is solved by blocking the paths to the goal vertex a bit before the goal vertex.

3.3. Construction. Place the goal vertex g at $(0, 0)$. For some sufficiently large integer $z \equiv 0(mod\ 10)$, define the outer rim R such that

$$R := \left\{(x, y) \in \mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0} : z^{0.7z} \leq x + y \leq z^{0.7z} + 1\right\}.$$

The rim is a long diagonal path from $(0, z^{0.7z})$ to about $(z^{0.7z}, 0)$. Note that the rim is two concentric quarter circles in the "taxicab" metric $\mathcal{L}_1$, so each point in R is within 1 of $z^{0.7z}$ from g. Along the rim, there will be "spoke-base points" and "column-base points" alternating. (Note: to avoid notational clutter, we omit the "floor" operation notation. Here, for example, $z^{0.7z}$ means $\lfloor z^{0.7z} \rfloor$.)

For each $i \in \{0.8z, \ldots, 0.9z\}$, create z^{z-i-1} spokes of class i. A conceptual figure is given in Figure 3. Let $S := \frac{z^{0.2z} - z^{0.1z-1}}{z-1}$ be the number of total spokes. For $i \in \{1, \ldots, S\}$, define the ith spoke-base, b_i, such that

$$b_i := \left(\frac{z^{0.7z}}{S} i + \frac{z^{0.7z}}{2S}, z^{0.7z} - \frac{z^{0.7z}}{S} i - \frac{z^{0.7z}}{2S}\right).$$

Therefore $b_i \in R$.

From each spoke-base, construct a twisty path towards g. Each path has length $2z^j$ for some $j \in \{0.8z, 0.8z+1, \ldots, 0.9z\}$. We call such a path of length $2z^j$ a "spoke of class j." The graph will contain z^{z-j-1} spokes of class j, for each j.

The taxicab distance between two adjacent spoke bases will be $2z^{0.7z}/S$, but to make the construction's key idea work, these distances must be longer for the robot as it moves in the graph. We increase the graph distances by inserting detour loops into the rim. Lemma 3.1 makes this idea precise.

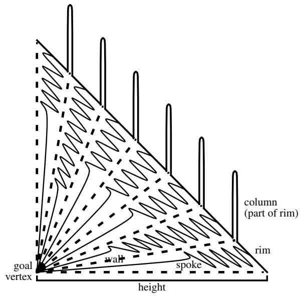


FIG. 3. *Conceptual figure with two spoke classes.*

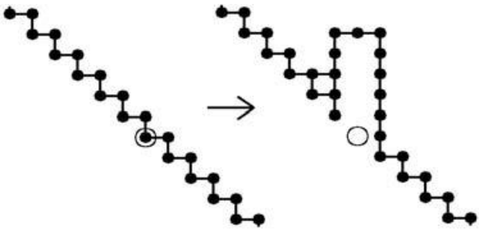


FIG. 4. *A column of height* 3.

LEMMA 3.1. *Given a function $t : \{1, \ldots, S\} \to \mathbb{Z}_{\geq 0}$ such that $t(i) - t(i-1) \geq \frac{2z^{0.7z}}{S+1}$ $\forall i$, it is possible to modify R using only cells above R in the plane, such that $\forall j > i$, traveling along R from b_i to b_j takes between $(t(j)-t(i)-4)$ and $(t(j)-t(i)+4)$ steps.*

Proof. For $i \in \{0, \ldots, S\}$, define the ith column-base, c_i such that

$$c_i := \left(\frac{z^{0.7z}}{S} i, z^{0.7z} - \frac{z^{0.7z}}{S} i \right).$$

Therefore $c_i \in R$. At each of the column-bases, remove the point itself and the point above it from the rim and add two paths traveling upwards, connected at the top. So, if the column-base is at (x, y), remove (x, y) and $(x, y+1)$ from R and add one path from $(x-1, y+2)$ to $(x-1, y+3+h)$ and another path from $(x+1, y)$ to $(x+1, y+3+h)$, with a connecting point at $(x, y+3+h)$. This increases the steps needed to cross this point in the rim by $2h$. We call such a construction a "column of height h." A column of height 3 is illustrated in Figure 4.

We now define an iterative algorithm for building the columns. For a fixed i, assume the previous columns have been built and let D be the current distance from b_1 to b_i. Let $h := \lfloor \frac{t(i)-t(1)-D}{2} \rfloor$ and build a column of height h at c_{i-1}. This ensures that the distance from b_1 to b_i is now $t(i) - t(1)$, up to round-off error. Repeat the process for all later i. (Note that the recalculation from $t(1)$ here prevents accumulation of round-off error.)

The fourth idea is to build each spoke as a twisty path of the appropriate length. Each spoke consists of a wedge of sufficient area. The spokes do not overlap, except in a small unblocked triangular region near the goal vertex, within which all paths are direct.

LEMMA 3.2. *Given a function $l : \{1, \ldots, S\} \to \mathbb{Z}_{\geq 0}$ such that $z^{0.7z} \leq l(i) \leq z^{z-1}$, $\forall i$, it is possible to construct spokes from b_i such that the distance from b_i to t is*

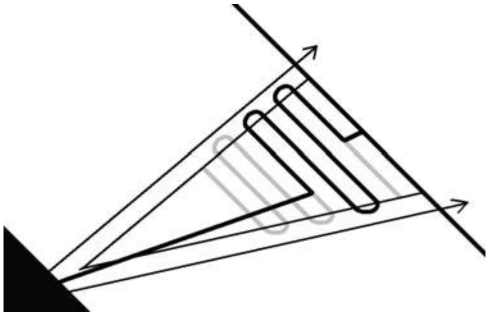


FIG. 5. *The hypothetical path P_i, along with the shortening necessary to set the path length to $l(i)$.*

between $(l(i) - 4)$ *and* $(l(i) + 4)$.

Proof. We would like to connect each of the spokes to g. However, the max degree of a grid prevents this. Therefore we add a triangle T such that

$$T := \left\{(x, y) : x, y \in \mathbb{Z}_{\geq 0} \bigwedge x + y \leq z^{0.5z}\right\}.$$

We may then simply connect the spokes to T. For each $i \in \{1, \dots, S\}$, define the ith "tip" vertex t_i such that

$$t_i := \left(\frac{z^{0.5z}}{S}i + \frac{z^{0.5z}}{2S}, z^{0.5z} - \frac{z^{0.5z}}{S}i - \frac{z^{0.5z}}{2S}\right).$$

Therefore $t_i \in T$. This will be the point that the ith spoke connects to.

To construct the paths of length $l(i)$ for each i, construct a hypothetical path P_i from b_i of excessive length. Then, when building the actual graph, simply include as much of P_i as necessary before the graph takes a direct path to t_i, as illustrated in Figure 5. The point where the graph ignores P_i and instead switches to a direct path to t_i depends on $l(i)$. We block the cell just prior to t_i on the path.

Building P_i takes a bit of construction. We use Euclidean rays from g to partition the space between R and g into areas A_i and then create many path segments running parallel to R called "levels." P_i runs up and down these levels, traveling back and forth to increase length. We define a space C_i to give room to connect one level to the next without coming close to the rays. This is all illustrated in Figure 5. The triangle in the lower left corner represents a region of unblocked cells, bordered by the t_i. Since all of the twisty paths have become direct paths by the time they reach their t_i, and the blockages occur prior to reaching t_i, the spokes may overlap within this unblocked region.

For each $i \in \{0, \dots, S\}$, define the ith "ray" r_i to be the Euclidean line from $(\frac{z^{0.5z}}{S}i, z^{0.5z} - \frac{z^{0.5z}}{S}i)$ to c_i. Hence r_i goes from T to R. For each $i \in \{1, \dots, S\}$, define the ith area A_i to be the integer points between r_{i-1} and r_i. Define the ith cushion C_i such that

$$C_i := \left\{(x, y) : x, y \in \mathbb{Z}_{\geq 0} \bigwedge d[((x, y), r_i)] \leq 8\right\}.$$

For each $i \in \{1, \dots, z^{0.3z} - 2\}$ and each $j \in \{1 \dots, 0.1z^{0.7z}\}$, define the level $l_{i,j}$ such that

$$l_{i,j} := \left\{(x, y) : z^{0.7z} - 6j \leq x + y \leq z^{0.7z} - 6j + 1\right\} \bigcap (A_i - C_i - C_{i+1}).$$

Use levels $\{l_{i,0}, \ldots, l_{i,0.1z^{0.7z}}\}$ to make P_i, using C_i and C_{i+1} to connect the levels. C_i is large enough to let P_i avoid crossing the ray.

The distance between c_i and c_{i+1} is $\frac{2z^{0.7z}}{S} \geq 2z^{0.3z}$. The levels are parallel and contained in a Euclidean triangle. The smallest is only one tenth of the way to the point, so each level is longer than $z^{0.3z}$. There are $0.1z^{0.7z}$ levels, so the total length of P_i is at least $0.1z^z$.

To build the actual spoke, we define a function $s(p)$ for all points $p \in P_i$, such that $s(p)$ is the distance from b_i to t if we were to shorten P_i at p and take a direct path to t_i from p. (Note that this definition involves the actual distance in the graph, avoiding accumulated round-off error.) Let S_i be the point in P_i that minimizes $|s(S_i) - l(i)|$. For any two points p, p' in the same level, if $d(p, p') = 2$, then $|s(p) - s(p')| = 2$, since $d(p, t) = d(p', t)$. Therefore, if S_i is contained in one of the levels, shortening P_i at S_i gives a spoke within 2 of $l(i)$. If S_i is contained in one of the cushions, we may be able to create an even more precise spoke. For any adjacent $c, c' \in C_i \cap P_i$, $|s(c) - s(c')| \leq 1$, as the path from c to t_i likely passes through c'. Hence, regardless of whether S_i appears in a level or in a cushion, we exceed the precision required by Lemma 3.2. □

To finally build our graph, define a function $p[i, j]$ such that

$$p[i, j] := z^{i+1} j + \frac{z^{0.8z+1}}{0.1z + 1}(i - 0.8z).$$

This will be the "position" of the jth spoke of class i. We order the spokes by this position function, so the first spoke is the one with the lowest position, the second is the one with the second lowest position, and so forth.

Put a blocked cell near the end of each spoke except one of class $0.9z$. Hence the robot will be tempted by each of the spokes of class $0.8z$ in turn, following the rim for about z^z steps. The robot will then turn around and travel up the rim, tempted only by the spokes of the next class $0.8z + 1$, again taking about z^z steps, and so on until it reaches the goal via the unblocked spoke of class $0.9z$.

Define the length function l such that $l[k] := 2z^i$, where i is the class of the kth spoke. We use this l with Lemma 3.1.

Define t such that $t[k] = p[i, j]$, where i and j are the coordinates for the kth spoke. We use this t with Lemma 3.2.

3.4. Analysis. Note: here we prove the lower bound for tactile (short-range) sensors. In section 5.1 we show that the theorem applies to all line-of-sight sensors as well.

THEOREM 3.3. *The worst-case travel distance of D^* on vertex-blocked grids $H = (V, E)$ is $\Omega(\frac{n \log n}{\log \log n})$ steps.*

Proof. The distance the robot must travel to find a spoke of class i and then travel to g is at most $z^{i+1} + 2z^i$ steps. For any $j > i$, simply traveling a spoke of class j will take at least $2z^j \geq 2z^{i+1}$ steps. Hence the robot will walk to the smallest class spoke available, find a blocked cell, and go to the next of that class, traversing the rim.

By the placement of the spokes, notice that $\forall h, h' \in \{0.8z, \ldots, 0.9z\}$, if $h < h'$, then the rightmost h-class spoke is to the right of the rightmost h'-class spoke. Also the leftmost h-class spoke is to the left of the leftmost h'-class spoke. Hence, after visiting every spoke of class i, the robot turns around and finds the first spoke of class $i + 1$. Each time the robot traverses the rim, it goes from the leftmost spoke of class

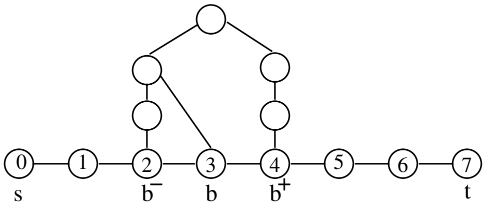


FIG. 6. *When the blockage at* b *of* P_i *is detected,* $d(2,t)$ *increases by at most* $d(b^-,b^+)^{H^{i+1}}-2=4$.

i to the rightmost spoke of class i. This distance is more than $z^z - 2z^{i-1} = \Omega(z^z)$. The total travel distance (just on the rim) is therefore at least $z^{0.1z}\Omega(z^z) = \Omega(z^{z+1})$.

On the other hand, there are $\theta(z^z)$ vertices in the rim (including the columns). There are $O(z^{0.5z})$ vertices in T. In class i there are $O(z^{z-i-1})$ spokes, each with $O(z^i)$ vertices, so each class contains $O(z^{z-1})$ vertices. There are $0.1z$ classes, so there are $O(z^{z-.9})$ vertices in the spokes. Therefore the total number of vertices in the graph is $\theta(z^z)$. If $n = \theta(z^z)$, then $\log n = \theta(z \log z)$ and $\log\log n = \theta(\log z)$. Then the total distance is $\Omega(z^{z+1}) = \Omega(nz) = \Omega(\frac{n \log n}{\log\log n})$. □

4. D^*: Upper bounds.

4.1. Notation.

As defined in section 2, the robot knows the graph $H = (V, E)$, the starting location $s \in V$ and a goal vertex $t \in V$. However, it does not know which vertices in V are blocked. D^* travels along a shortest presumed unblocked path to t. If the robot has tactile sensors, it replans whenever it encounters a blocked vertex along its currently planned path. To prepare the way for an extension to long-range sensors in the next section, we analyze here a slightly more general case. We permit the robot to detect a blocked vertex some distance ahead on its planned path. For example, in Figure 6, the robot starting from 0 might travel as far as 2 and then detect blocked vertex 6. Note that an earlier vertex such as 4 might be blocked, but go undetected at this iteration.

We assume that the initial graph $H = (V, E)$ given to the robot is connected with $n = |V|$ vertices (if not, take the component containing the starting vertex). The starting and target vertices are denoted $s, t \in V$, respectively. At the start of the ith iteration of D^*, let v_{i-1} denote the robot's location and $H^i = (V, E_i)$ denote its current information about the environment. E_i is obtained from E by removing all edges incident on vertices that have been found to be blocked. Initially $v_0 = s$ and $H^1 = H$. Let P_i denote the shortest path in H^i from v_{i-1} to t that the robot decides to follow. If i is not the final iteration, let b_i be the vertex found to be blocked by the robot while following P_i. H^{i+1} is obtained from H^i by removing edges incident on b_i. Let b_i^- and b_i^+ denote, respectively, the vertices preceding and following b_i on P_i; see Figure 6. Let v_i be the starting vertex for the next iteration. Clearly v_i either is b_i^- in P_i or precedes b_i^- in P_i and the subpath of P_i between v_i and b_i^- exists in H^{i+1}. Also, the subpath of P_i from b_i^+ to t exists in H^{i+1}.

Let $d(u, v)^H$ denote the shortest distance between vertices u and v in graph H. If u and v are not connected, then $d(u, v)^H = \infty$.

Let $v_0, v_1, \ldots, v_k$ be a run of the method. This captures a run up to ties in shortest viable paths. If the robot reaches t, then $v_k = t$. The total distance traveled by the robot is

$$C = \sum_{i=1}^{k} d(v_{i-1}, v_i)^{H^i}.$$

4.2. Telescoping.

LEMMA 4.1. $C \leq n + \sum_{i=1}^{k-1} d(b_i^-, b_i^+)^{H^{i+1}}$.

Proof. Since v_i lies on the shortest path P_i from v_{i-1} to t in H^i, by the principle of optimality

$$\begin{aligned} C = \sum_{i=1}^{k} d(v_{i-1}, v_i)^{H^i} &= \sum_{i=1}^{k} (d(v_{i-1}, t)^{H^i} - d(v_i, t)^{H^i}) \\ = d(v_0, t)^{H^1} - d(v_k, t)^{H^k} &+ \sum_{i=1}^{k-1} (d(v_i, t)^{H^{i+1}} - d(v_i, t)^{H^i}) \\ &\leq n + \sum_{i=1}^{k-1} (d(v_i, t)^{H^{i+1}} - d(v_i, t)^{H^i}). \end{aligned}$$

This formula has the following intuitive explanation: the robot optimistically thinks that undetected vertices are unblocked. When the robot gets to v_i and detects a blockage, it is set back in the distance it thinks it is from t, by the amount $(d(v_i, t)^{H^{i+1}} - d(v_i, t)^{H^i})$. The sum of these setbacks, plus the initial optimistic distance to t, equals the total distance traveled by the robot.

By the triangle inequality,

$$d(v_i, t)^{H^{i+1}} \leq d(v_i, b_i^-)^{H^{i+1}} + d(b_i^-, b_i^+)^{H^{i+1}} + d(b_i^+, t)^{H^{i+1}}. \tag{4.1}$$

By the principle of optimality, the subpath of P_i from v_i to b_i^- in H^i has length $d(v_i, b_i^-)^{H^i}$, the subpath of P_i from b_i^- to b_i^+ has length $d(b_i^-, b_i^+)^{H^i} = 2$, and the subpath of P_i from b_i^+ to t in H^i has length $d(b_i^+, t)^{H^i}$. Hence,

$$d(v_i, t)^{H^i} = d(v_i, b_i^-)^{H^i} + 2 + d(b_i^+, t)^{H^i}. \tag{4.2}$$

Observe that the first and third of these subpaths exist in H^{i+1}. Only the path of length 2 through b_i between b_i^- and b_i^+ is no longer viable in H^{i+1}. Therefore, $d(v_i, b_i^-)^{H^i} = d(v_i, b_i^-)^{H^{i+1}}$ and $d(b_i^+, t)^{H^i} = d(b_i^+, t)^{H^{i+1}}$. Plugging (4.1) and (4.2) into the bound for C above yields the lemma. ∎

In plain words, the amount of the setback when at v_i cannot be more than the revised distance $d(b_i^-, b_i^+)^{H^{i+1}} - 2$ since the robot could splice in that path to replace the blocked b_i^-, b_i, b_i^+ portion of P_i. Notice that $d(b_i^-, b_i^+)^{H^{i+1}} < \infty$ because the following pairs are all in the same connected component in H^{i+1}: v_i and b_i^-; v_i and t; b_i^+ and t.

4.3. Time reversal and weighted edges. Define the following function:

CYCLE-WEIGHT(T, S). *Input*: a tree $T = (V, F)$ and an ordered list $S = \{e_k, e_{k-1}, \ldots, e_1\}$ of distinct edges from the complete graph on V such that $S \cap F = \phi$. Define the weight w_i of edge $e_i \in S$ to be the length of a shortest cycle that contains e_i in the graph $T_i = (V, F \cup \{e_k, e_{k-1}, \ldots, e_i\})$.
Output: $\sum_{i=1}^{k} w_i$.

We next show that $\sum_{i=1}^{k-1} d(b_i^-, b_i^+)^{H^{i+1}} \leq$ CYCLE-WEIGHT(T, S) for a suitably constructed tree T and $S = \{e_i = (b_i^-, b_i) : 1 \leq i \leq k-1\}$.

The basic idea relating the edge weights in CYCLE-WEIGHT to the $d(b_i^-, b_i^+)^{H^{i+1}}$ values can be understood by considering a special case. Suppose H^k is connected except for the isolated vertices $b_1, b_2, \ldots, b_{k-1}$. Reverse the time perspective so that the robot motion *adds* edges, first the edges incident on b_{k-1}, then the edges incident on b_{k-2}, and so on. Pick T to be a spanning tree of the graph $(V, E_k \cup \{(b_1, b_1^+), (b_2, b_2^+), \ldots, (b_{k-1}, b_{k-1}^+)\})$ and S to be $e_i = (b_i^-, b_i) : 1 \leq i \leq k-1$. Then $w_i \geq 2 + d(b_i^-, b_i^+)^{H^{i+1}}$ because any cycle containing (b_i^-, b_i) in T_i must also contain (b_i, b_i^+).

Unfortunately such a simple construction does not work in the general case as multiple connected components may be formed when the edges incident to a blocked vertex are removed. To get around this problem, we define a new sequence of graphs $F_k, F_{k-1}, \ldots, F_1$ as follows:

1. F_k is a spanning forest of H^k.
2. For $1 \leq i \leq k-1$, let C^i be the connected component of H^{i+1} containing b_i^+ and b_i^-. Then F_i is a spanning forest of H^i containing the subgraph $F_{i+1} \bigcup \{(b_i, b_i^+)\}$.

The following lemma follows by induction directly from the definition of F_i.

LEMMA 4.2. *For $1 \leq i \leq k$ and all vertices u and v, F_i is acyclic; $d(u,v)^{F_i} < \infty$ iff $d(u,v)^{H_i} < \infty$; and $d(u,v)^{F_i} \geq d(u,v)^{H^i}$.*

Consider the cycle weight problem with $T = F_1$ and $S = \{e_i = (b_i, b_i^-) : 1 \leq i \leq k-1\}$. The next lemma bounds the cost of our method by CYCLE-WEIGHT(T,S).

LEMMA 4.3. *Let $H^1, H^2, \ldots, H^k$ be a sequence of graphs as defined in section 4.1. Let $T = F_1$ and $S = \{e_i = (b_i^-, b_i) : 1 \leq i \leq k-1\}$. Then $\sum_{i=1}^{k-1} d(b_i^-, b_i^+)^{H^{i+1}} \leq$ CYCLE-WEIGHT(T,S).*

Proof. According to Lemma 4.2, F_{i+1} and H^{i+1} have the same connected components. The subgraph of F_1 induced by C^i is connected since C^i is a component of H^{i+1}. The edges e_j for $i < j < k$ are contained in C^i since $b_j^-, b_j, b_j^+ \in C^i$ for all $i < j < k$. Thus, the graph obtained by contracting all vertices of C^i in T_{i+1} is acyclic. Since T_i is obtained from T_{i+1} by adding e_i, every cycle that contains $e_i = (b_i^-, b_i)$ in T_i must also contain (b_i, b_i^+). Thus, w_i is equal to 2 plus the distance between b_i^- and b_i^+ in the subgraph G' of T_i induced by C^i. But G' is also a subgraph of H^{i+1} and hence it holds that $w_i \geq 2 + d(b_i^-, b_i^+)^{H^{i+1}}$. Consequently, $\sum_{i=1}^{k-1} d(b_i^-, b_i^+)^{H^{i+1}} \leq \sum_{i=1}^{k-1} w_i =$ CYCLE-WEIGHT(T, S). □

4.4. An extremal problem on graphs. We now bound CYCLE-WEIGHT$((V, E), S)$ in terms of $|V|$ and $|S|$. Let $E_w = \{e_i; w_i \geq w\}$ be the set of edges with weight at least w. Recall that the *girth* of a graph is the length of its shortest cycle. Define $\Gamma(n, w)$ (respectively, $\Gamma_P(n, w)$) to denote the maximum number of edges in a graph (respectively, planar graph) with n vertices and a girth of at least w. The following lemma relates E_w and $\Gamma(n, w)$.

LEMMA 4.4. $|E_w| \leq \Gamma(|V|, w) - |V| + 1$ *for all CYCLE-WEIGHT$((V, E), S)$ and all w.*

Proof. Consider the graph $T_w = (V, E \cup E_w)$. We claim that T_w has a girth of at least w. To see this, assume that it does not and thus has a cycle C of length $w' < w$. Since (V, E) is a tree, at least one edge of C must belong to E_w. Consider the edge $e_j \in E_w \cap C$ with the smallest j. Then T_j contains C and thus $w_j \leq w' < w$. On the other hand, $w_j \geq w$ since $e_j \in E_w$, which is a contradiction. Thus, T_w has a girth of at least w. This implies that $\Gamma(|V|, w) \geq |E \cup E_w| = |E| + |E_w| = |V| - 1 + |E_w|$ and the lemma follows. □

Corollary 4.5. *$|E_w| \leq \Gamma_P(|V|, w) - |V| + 1$ for all CYCLE-WEIGHT$((V, E), S)$ such that $(V, E \cup S)$ is planar, and all w.*

Proof. In the proof of Lemma 4.4, T_w is planar because it is a subgraph of planar graph $(V, E \cup S)$. Hence $\Gamma(|V|, w)$ may be replaced by $\Gamma_P(|V|, w)$. □

We now bound CYCLE-WEIGHT$((V, E), S)$ by making use of bounds on $\Gamma(n, w)$, a well studied problem in extremal combinatorics. We first consider the case that the graph $(V, E \cup S)$ is planar.

Lemma 4.6. *$\Gamma_P(n, w) \leq \frac{wn}{w-2}$ for all n and w.*

Proof. Since the sum of the lengths of all faces of any planar graph $G = (V, E)$ is at most $2|E|$ and every face has length at least w, the number of its faces can be at most $2|E|/w$. The bound of the lemma follows from substituting this relationship in Euler's formula. □

Note that the weight of any edge in S is at most $|V|$. Define $E_{w,2w} = \{e_i \in S : w \leq w_i < 2w\}$. Then, by Corollary 4.5 and Lemma 4.6 it holds that

$$\begin{aligned}
\text{CYCLE-WEIGHT}((V,E),S) &\leq \sum_{i=1}^{\log|V|} 2^{i+1}|E_{2^i,2^{i+1}}| \\
&\leq O(|S|) + \sum_{i=3}^{\log|V|} 2^{i+1}|E_{2^i}| \\
&\leq O(|S|) + \sum_{i=3}^{\log|V|} 2^{i+1}(\Gamma_P(|V|, 2^i) - |V| + 1) \\
&\leq O(|S|) + \sum_{i=3}^{\log|V|} 2^{i+1}\left(\frac{2^i|V|}{2^i-2} - |V| + 1\right) \\
&\leq O(|S|) + \sum_{i=3}^{\log|V|} 2^{i+1}\, 4|V|/2^i \\
&= O(|V|\log|V|).
\end{aligned}$$

The last inequality depends on planarity (so $S = O(|V|)$) and $|V| \geq 6$. We now repeat the analysis for general graphs. In this case, we use a recent result by Alon, Hoory, and Linial [1] that states that any graph $G = (V, E)$ with average degree $d > 2$ has a girth of at most $\log_{d-1}|V|$ [1], resulting in the following lemma.

Lemma 4.7. *$\Gamma(n, w) \leq n(n^{\frac{1}{w}} + 1)/2$ for all n and w.*

Proof. Consider any graph $G = (V, E)$ with $|V| = n$, $|E| \geq |V| + 1$ and a girth of at least w. Then, its average degree is $d = 2|E|/n > 2$ and thus, according to the result by Alon, Hoory, and Linial [1], $w \leq \log_{2|E|/n-1} n$. Solving this inequality for $|E|$ yields the lemma. □

This lemma allows us to bound CYCLE-WEIGHT$((V, E), S)$ for general graphs.

Lemma 4.8. *$w(|V|(|V|^{\frac{1}{w}} - 1)) = O(|V|\log|V|)$ for $|V| \geq w > \log^2|V|$.*

Proof. Let $n = |V|$ and remove the common factor $|V|$ from the statement of the lemma. The resulting left-hand side defines the function $f(w) = w(n^{\frac{1}{w}} - 1)$. Its derivative is

$$f'(w) = n^{\frac{1}{w}}\left(1 - \frac{\ln n}{w}\right) - 1$$

and its second derivative is

$$f''(w) = \frac{n^{\frac{1}{w}} \ln^2 n}{w^3} > 0.$$

Therefore f is convex (in the range $w > 0$). Hence $\arg\max_{n \geq w \geq \log^2 n} f(w)$ occurs at one of the endpoints of the range, n or $\log^2 n$. We will show that $f(w) = O(\log n)$ for both endpoints.

At $w = n$, let $t = \frac{\ln n}{n} \to 0$ as $n \to \infty$. The Taylor series for e^t around 0 then gives

$$n^{\frac{1}{n}} - 1 = e^{\frac{\ln n}{n}} - 1 = e^t - 1 = \frac{\ln n}{n} + \frac{ln^2 n}{2n^2} + o(n^{-2}) = O\left(\frac{\log n}{n}\right).$$

Thus $f(n) = O(\log n)$.

At $w = \log^2 n$, let $t = \frac{\ln 2}{\log n}$, so

$$\frac{f(w)}{\log n} = \log n (n^{\frac{1}{\log^2 n}}) - 1 = \log n (e^{\frac{\ln n}{\log^2 n}} - 1) = \log n (e^{\frac{\ln 2}{\log n}} - 1) = \frac{\ln 2}{t}(e^t - 1).$$

Again using the Taylor series we get $\frac{f(w)}{\log n} = \ln 2(1 + \frac{t}{2} + \frac{t^2}{6} + \cdots) = \ln 2(1 + o(1)) = O(1)$. □

Using Lemmata 4.8, 4.4, and 4.7, we have

$$\begin{aligned}
\text{CYCLE-WEIGHT}((V,E),S) &= \sum_{i: w_i \leq \log^2 |V|} w_i + \sum_{i: w_i > \log^2 |V|} w_i \\
&\leq |S| \log^2 |V| + \sum_{i=2\log\log|V|}^{\log|V|} 2^{i+1} |E_{2^i, 2^{i+1}}| \\
&\leq |S| \log^2 |V| + \sum_{i=2\log\log|V|}^{\log|V|} 2^{i+1} |E_{2^i}| \\
&\leq |S| \log^2 |V| + \sum_{i=2\log\log|V|}^{\log|V|} 2^{i+1} (\Gamma(|V|, 2^i) - |V| + 1) \\
&= |S| \log^2 |V| + \sum_{i=2\log\log|V|}^{\log|V|} 2^{i+1} (|V|(|V|^{\frac{1}{2^i}} - 1)/2 + 1) \\
&= |S| \log^2 |V| + \sum_{i=2\log\log|V|}^{\log|V|} O(|V| \log |V|) \\
&= O((|V| + |S|) \log^2 |V|).
\end{aligned}$$

We now state these results as a lemma.

LEMMA 4.9. *CYCLE-WEIGHT*$((V,E),S) = O((|V| + |S|)\log^2 |V|)$. *If the graph* $(V, E \cup S)$ *is planar, CYCLE-WEIGHT*$((V,E),S) = O(|V| \log |V|)$.

4.5. Worst-case travel bound. We are now ready to prove an upper bound on the worst-case travel distance of D^*.

Theorem 4.10. *For robot sensors as described in section* 4.1*, D^* traverses $O(n \log^2 n)$ edges on connected graphs $G = (V, E)$. It traverses $O(n \log n)$ edges on connected planar graphs $G = (V, E)$.*

Proof. According to Lemmata 4.1 and 4.3, D^* traverses at most $O(n) + \sum_{i=1}^{k-1} d(b_i^-, b_i^+)^{H^{i+1}} \leq O(n) + \text{CYCLE-WEIGHT}((V, E'), S)$ edges, where $|S| < n$ and $(V, E' \cup S)$ is a subgraph of G. According to Lemma 4.9, it holds that $\text{CYCLE-WEIGHT}((V, E'), S) = O((n + |S|) \log^2 n) = O(n \log^2 n)$ and, if G and thus $(V, E' \cup S)$ are planar, $\text{CYCLE-WEIGHT}((V, E'), S) = O(n \log n)$. □

5. Extensions.

5.1. Long-range sensors.

Both the lower and upper bounds of the previous sections extend to the case of long-range sensors, rather than the tactile sensors we have assumed so far. Many real robots are equipped with sonar, radar, or laser sensors, so it is worthwhile to consider this case. In directions where the view is not blocked by obstacles, these sensors can detect at moderate or even unlimited distances.

The lower bound is easy. Place a little twist in the path P_i just before the blocked vertex of each spoke, so that the blocked vertex cannot be detected until the robot is $O(1)$ vertices away. Therefore Theorem 3.3 applies to robots with long-range field-of-vision sensors.

We now extend the upper bound to the case of long-range sensors. We will not require that the sensors be field-of-vision; they may see around corners, have gaps in their vision, etc. We only require that if the robot attempts to move to vertex $v \in B$ from a vertex adjacent to v, then the robot will detect that $v \in B$. This is a minimal property required for any functioning robot.

Theorem 5.1. *Suppose that the robot follows the D^* algorithm on graph $H = (V, E)$. Each time the robot attempts to move to an adjacent vertex, it either moves successfully or it detects that the vertex is blocked. After an attempted move (whether successful or not) the robot may detect additional blocked vertices in H. Then the bounds of Theorem* 4.10 *apply.*

Proof. Our proof consists of two parts. Part 1 shows that our bounds apply if the robot detects blocked vertices that are not on the planned path to the target. Part 2 shows that if more than one blocked vertex on the planned path is detected, then there exists a different robot whose movements are the same, but which does not detect more than one blocked vertex on the planned path.

We preface part 1 by stating the very simple ideas hidden in the technical statements. Blocked vertices off the path do not affect the telescoping formula of Lemma 4.1, because, by definition, they do not affect the current path. When we reverse time and add the special edges $e_k, \ldots, e_1$, we add extra edges (those connected to the off-path vertices). Our upper bound is on the length of a smallest cycle containing e_i, so adding extra edges can only make this smaller. Therefore the upper bound, which is computed in Lemma 4.9 as though there were no extra edges, is still valid.

Let $B_i \subset V$ denote the off-path vertices detected as blocked in iteration i. The definitions of v_i and H^i remain the same as in section 4.1, but now H^{i+1} is obtained from H^i by removing all edges incident on b_i or incident on any $b \in B_i$. Lemma 4.1 remains true in this setting because no vertices in B_i are on the path P_i. In particular, the subpaths of P_i from v_i to b_i^- and from b_i^+ to t still exist in H^{i+1}. Intuitively, the blockages B_i contribute to the setback amount suffered by the robot, but this setback is still bounded by the change in distance from b_i^- to b_i^+.

For the associated cycle weight problem, we define a sequence of forests

$F_k, F_{k-1}, \ldots, F_1$. As before, F_k is a spanning forest of H_k and F_i is a spanning forest of H_i containing the subgraph $F_{i+1} \bigcup \{(b_i, b_i^+)\}$. It is easy to show that taking $T = F_1$ and $S = \{e_i = (b_i, b_i^-) : 1 \leq i \leq k-1\}$ satisfies Lemma 4.3. Therefore we have verified part 1.

Based on part 1, the bounds of Theorem 4.10 apply as long as the robot never detects more than one blocked vertex on the current planned path to t. For the second part of the proof, whenever the robot detects more than one such blocked vertex, categorize the detected vertices as follows:

- *off-path:* all vertices not on the current planned (shortest presumed unblocked) path to t.
- *first-path:* the nearest detected blocked vertex on the current planned path to t.
- *more-path:* all other detected blocked vertices on the current planned path to t.

Consider now a fictional robot whose movements have been identical to the real one, and which until the present step has detected the same set of blocked vertices. Now, however, our fictional robot only detects the off-path vertices and first-path vertex. It replans the shortest presumed unblocked path to t, moves zero steps, and then considers detecting the more-path vertices (more-path with respect to the original plan, not the new plan). It detects all of those which are off the newly replanned path. It can also detect one vertex on the new path, if there is one. If more than one of these are on the new path, it recategorizes them with respect to the new path and repeats the procedure.

This procedure must terminate, because each replan strictly decreases the number of more-path vertices. At termination, the fictional robot has performed precisely the same set of physical movements as has the real robot, and it has detected the same set of blocked vertices. The fictional robot has never detected more than one blocked vertex on its current planned path. The desired bounds therefore apply to both it and the real robot. □

5.2. Blocked edges. Another natural extension is when in addition to blocked vertices $B \subset V$, some edges $B' \subset E$ might also be blocked. This can be reduced to the vertex blocking case by adding a new vertex v_e in the middle of every edge $e \in E$. Blocking of e then corresponds to blocking of vertex v_e in the tranformed graph. We consider two cases.

First, assume that the robot does not expend travel cost to detect an incident blocked edge. Then if the robot encounters a blocked edge (u, v) while going from u to v, it can sense all other edges emanating from u to check which ones are blocked at zero additional cost. Thus the robot will stop in at most n iterations. To bound the travel cost, let (b_i^-, b_i^+) be the edge found blocked by the robot in iteration i. Lemma 4.1 remains true in this setting as the subpaths of P_i from v_i to b_i^- and from b_i^+ to t still exist in H^{i+1}. For the associated cycle weight problem, H^{i+1} is now obtained from H^i by removing all edges found blocked by the robot in iteration i. Define the sequence $F_k, F_{k-1}, \ldots, F_1$ by taking F_k a spanning forest of H^k and F_i a spanning forest of H^i containing the subgraph F_{i+1}. Similar arguments show that $T = F_1, S = \{(b_i^-, b_i^+) : 1 \leq i \leq k-1\}$ satisfies Lemma 4.3. By arguments for long-range sensors above, the bounds in Theorem 4.10 also hold when the robot detects a combination of blocked vertices and edges in each iteration.

Next we assume that the robot must traverse an edge in order to detect edge blockage. In this case detecting blocked e in the original graph corresponds to traveling to vertex v_e in the transformed graph. However, the number of vertices in the transformed graph is $|V| + |E|$ and Theorem 4.10 gives an $O(|E| \log |E|)$ upper bound

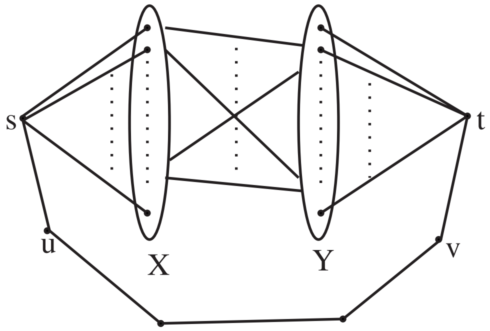


FIG. 7. *Lower bound example for blocked edges.*

for planar graphs and $O(|E|\log^2|E|)$ upper bound for general graphs. For planar graphs this is still $O(n\log n)$ since $|E| = O(n)$. We next show a lower bound of $\Omega(|E|)$ for D^* on general graphs. Thus our bounds leave a $O(\log^2|E|) = O(\log^2 n)$ gap.

Consider the graph $H = (\{s,t\}\bigcup X\bigcup Y, E)$ as shown in Figure 7 where $|X| = |Y| = \frac{n}{2}$. Assume that all edges $E' \subset E$ between X and Y are blocked without the knowledge of the robot. Imagine a little twist towards the end of each edge $e \in E'$, so the robot has to travel to the twist to find out whether e is blocked. Now consider running D^* with start vertex s and target vertex t. As long as there exists a "presumed unblocked" edge $(x,y) \in E'$ at the start of iteration i, the robot has a length 2 path $v_{i-1} - y - t$ or a length 4 path $v_{i-1} - s - x - y - t$ available to it. Therefore the robot will not take the length 6 path $v_{i-1} - s - u - \cdots - v - t$ until iteration $|E'|+1$. In each preceding iteration, the robot will travel on edge (x,y) till the twist near y, find it to be blocked, and then come back to x. Therefore its travel cost is at least $\Omega(|E'|) = \Omega(|E|)$ steps on H.

6. Conclusions. The popular robot-navigation method that we have analyzed in this paper, D^*, is appealingly simple and easy to implement from a robotics point of view and appealingly complicated to analyze from a mathematical point of view. Our results, likewise, are satisfying in two ways. First, our tighter upper bounds on worst-case travel distances guarantee that D^* cannot perform badly under any circumstances. Second, the gap between the best known lower and upper bounds is now quite small, namely $O(\log\log n)$ for planar graphs, and $O(\log n\log\log n)$ on arbitrary graphs.

REFERENCES

[1] N. Alon, S. Hoory, and N. Linial, *The Moore bound for irregular graphs*, Graphs Combin., 18 (2002), pp. 53–57.

[2] B. Brumitt and A. Stentz, *GRAMMPS: A generalized mission planner for multiple mobile robots*, in Proceedings of the International Conference on Robotics and Automation, Leuven, Belgium, 1998.

[3] M. Hebert, R. McLachlan, and P. Chang, *Experiments with driving modes for urban robots*, in Proceedings of the SPIE Mobile Robots, Boston, 1999.

[4] S. Koenig and M. Likhachev, *Improved fast replanning for robot navigation in unknown terrain*, in Proceedings of the International Conference on Robotics and Automation, Washington, D.C., 2002, pp. 968–975.

[5] S. Koenig, C. Tovey, and W. Halliburton, *Greedy mapping of terrain*, in Proceedings of the International Conference on Robotics and Automation, Seoul, Korea, 2001, pp. 3594–3599.

[6] S. Koenig, C. Tovey, and Y. Smirnov, *Performance bounds for planning in unknown terrain. Planning with uncertainty and incomplete information*, Artificial Intelligence, 147 (2003), pp. 253–279.

[7] L. Matthies, Y. Xiong, R. Hogg, D. Zhu, A. Rankin, B. Kennedy, M. Hebert, R. Maclachlan, C. Won, T. Frost, G. Sukhatme, M. McHenry, and S. Goldberg, *A portable, autonomous, urban reconnaissance robot*, in Proceedings of the International Conference on Intelligent Autonomous Systems, Paris, France, 2000.

[8] I. Nourbakhsh, *Interleaving Planning and Execution for Autonomous Robots*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.

[9] I. Nourbakhsh and M. Genesereth, *Assumptive planning and execution: a simple, working robot architecture*, Autonomous Robots Journal, 3 (1996), pp. 49–67.

[10] Y. Smirnov, *Hybrid Algorithms for On-Line Search and Combinatorial Optimization Problems*, Tech. report CMU-CS-97-171, Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, 1997; available online from http://www.cs.cmu.edu/afs/cs/cmu.edu/user/smir/www/home/html.

[11] A. Stentz, *The focused D* algorithm for real-time replanning*, in Proceedings of the International Joint Conference on Artificial Intelligence, Montreal, Quebec, 1995, pp. 1652–1659.

[12] A. Stentz, *CD*: A real-time resolution optimal re-planner for globally constrained problems*, in Proceedings of the National Conference on Artificial Intelligence, Edmonton, Alberta, 2002, pp. 605–612.

[13] A. Stentz and M. Hebert, *A complete navigation system for goal acquisition in unknown environments*, Autonomous Robots, 2 (1995), pp. 127–145.

[14] S. Thayer, B. Digney, M. Diaz, A. Stentz, B. Nabbe, and M. Hebert, *Distributed robotic mapping of extreme environments*, in Proceedings of the SPIE: Mobile Robots XV and Telemanipulator and Telepresence Technologies VII, Vol. 4195, Boston, 2000.

SIAM J. DISCRETE MATH.
Vol. 19, No. 2, pp. 448–462

LABELING SCHEMES FOR SMALL DISTANCES IN TREES[*]

STEPHEN ALSTRUP[†], PHILIP BILLE[†], AND THEIS RAUHE[†]

Abstract. We consider labeling schemes for trees, supporting various relationships between nodes at small distance. For instance, we show that given a tree T and an integer k we can assign labels to each node of T such that given the label of two nodes we can decide, from these two labels alone, if the distance between v and w is at most k and, if so, compute it. For trees with n nodes and $k \geq 2$, we give a lower bound on the maximum label length of $\log n + \Omega(\log \log n)$ bits, and for constant k, we give an upper bound of $\log n + O(\log \log n)$. Bounds for ancestor, sibling, connectivity, and bi- and triconnectivity labeling schemes are also presented.

Key words. labeling schemes, trees

AMS subject classifications. 68R10, 68W01

DOI. 10.1137/S0895480103433409

1. Introduction.

Motivated by applications in XML search engines, network routing, and implicit graph representation, several *labeling schemes* for trees have been developed, among these [16, 22, 13, 10, 26, 1, 3, 8]. Given a tree, a labeling scheme assigns a *label*, which is a binary string, to each node v of the tree. Then, given only the labels of two nodes we can compute some predefined function of the two nodes. The main objective is to minimize the *maximum label length*, i.e., the maximum number of bits used in a label.

In this paper we consider labeling schemes for various relationships between nodes of small distance in trees. For instance, we show, by giving upper and lower bounds, that a labeling scheme supporting parent and sibling queries requires labels of length $\log n + \Theta(\log \log n)$.[1] This improves a recent bound by Kaplan and Milo [18] of $\log n + O(\sqrt{\log n})$.

More generally, we say that two nodes v and w with nearest common ancestor z are (k_1, k_2)*-related* if the distance from v to z is k_1 and the distance from w to z is k_2. For a positive integer k, a k*-relationship labeling scheme* is a labeling scheme for trees which supports tests for whether v and w are (k_1, k_2)-related for all nodes v and w and all positive integers $k_1, k_2 \leq k$. In particular, a 1-relationship labeling scheme supports tests for whether two nodes are $(0,0)$-, $(0,1)$-, $(1,0)$-, or $(1,1)$-related, that is, whether two nodes are identical, one is the parent of the other, or they are siblings. For trees with n nodes we show, for $k = 1$, a lower bound on the label length of $\log n + \Omega(\log \log n)$, and for fixed, constant k we give an upper bound of $\log n + O(\log \log n)$.

As noted in [18], a k-relationship labeling scheme can be used to test whether the distance between two nodes is at most k, and if this is the case we can compute the distance exactly. We call a labeling scheme with this property a k*-restricted distance labeling scheme*. We give a lower bound showing that for $k = 2$, a k-restricted

[*] Received by the editors August 15, 2003; accepted for publication (in revised form) November 29, 2004; published electronically November 4, 2005. An extended abstract of this paper appeared in *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2003.
http://www.siam.org/journals/sidma/19-2/43340.html

[†] IT University of Copenhagen, Rued Langgardsvej 7, DK-2300 Copenhagen S, Denmark (stephen@itu.dk, beetle@itu.dk, theis@itu.dk).

[1] log refers to the binary logarithm and $\log^*$ is the number of times log should be iterated to get a constant.

distance labeling scheme requires labels of length $\log n + \Omega(\log\log n)$. Hence, for constant k, our k-relationship labeling scheme gives a k-restricted distance labeling scheme which is optimal to within a factor of $\log\log n$. This result improves a recent upper bound of $\log n + O(\sqrt{\log n})$ for k-relationship and k-restricted distance labeling schemes given in [18]. In contrast to the results for restricted distances, Gavoille et al. [13] show that a labeling scheme for computing the distance between any pair of nodes in a tree must use labels of length $\Theta(\log^2 n)$. In [10] it is shown that even if the distances are allowed to be approximated to within a factor of $(1 + 1/\log n)$ we still need labels of length $\Theta(\log n \log\log n)$. Our result shows that for restricted distances much smaller labels suffice. A 1-restricted labeling scheme supports tests for whether two nodes are identical or adjacent. Such a labeling scheme, called an *adjacency labeling scheme*, was recently given for trees in [4], with label length bounded by $\log n + O(\log^* n)$. Thus, there is a provable gap between the label length of 1- and 2-restricted distance labeling schemes.

The above lower bounds are the result of a more general new technique which we use to obtain lower bounds for several types of labeling schemes, and for many of these we give matching upper bounds. Apart from the above results we present the following.

1.1. Bi- and triconnectivity labeling schemes. As an application of our k-relationship labeling scheme we obtain a labeling scheme for general graphs for biconnectivity (or 2-vertex connectivity) queries. Recently, Katz et al. [21] considered labeling schemes for 1-, 2-, 3-, and m-vertex connectivity. They gave a labeling scheme for biconnectivity using $3 \log n$ bits. We show, giving upper and lower bounds, that labels of length $\log n + \Theta(\log\log n)$ are required. The labeling scheme for triconnectivity (or 3-vertex connectivity) in [21] uses the biconnectivity labeling scheme and has label length bounded by $5 \log n$. Using our biconnectivity labeling scheme we obtain a triconnectivity labeling scheme using labels of length $3 \log n + O(\log\log n)$.

1.2. Ancestor labeling schemes. For trees with n nodes we show that a labeling scheme for ancestor queries must use labels of length $\log n + \Omega(\log\log n)$. This is the first nontrivial lower bound for the problem. Upper bounds using $2\lceil\log n\rceil$ bits were given in [27, 17, 23]. Recently, Abiteboul, Kaplan, and Milo [1] gave an ancestor labeling scheme using labels of length $3/2 \log n + O(\log\log n)$. Subsequently, this was improved by Alstrup and Rauhe [3], bounding the label length to $\log n + O(\sqrt{\log n})$.

If no two nodes are assigned to the same label, we say that the labels are unique. The above labeling schemes all produce unique labels, whereas the lower bounds also hold for labeling schemes that produce nonunique labels. However, the following bounds show that there is a nontrivial complexity difference between labeling schemes assigning unique and nonunique labels.

1.3. Sibling and connectivity labeling schemes. For sibling queries we give a labeling scheme using labels of length $\lceil\log n\rceil$. This labeling scheme will not assign unique labels to the nodes of the tree. For the case where uniqueness is required, as in [16], we give upper and lower bounds of $\log n + \Theta(\log\log \Delta)$ for trees of maximum degree Δ. Extending the result for the sibling labeling scheme we give a labeling scheme supporting connectivity queries for forests of n nodes using labels of length $\lceil\log n\rceil$. Again, these labels are not unique, and if uniqueness is required we show that labels of length $\log n + \Theta(\log\log n)$ are required.

1.4. Related work. Adjacency labeling schemes were introduced by Breuer and Folkman [5, 6], and efficient labeling schemes were considered by Kannan, Naor, and

Rudich in [16, 17]. In [22] *distance labeling schemes* were introduced, i.e., labeling schemes that compute the distance between any pair of nodes. Distance labeling schemes for various types of graphs are given in [22, 20, 13, 11], and distance labeling schemes computing approximate distances are given in [10, 25].

Recently, labeling schemes for various other relationships have been studied. Labeling schemes are given for ancestor in [17, 1, 26, 3, 19, 8], for nearest common ancestor in [2], and for connectivity in [21]. Efficient labeling schemes are also applicable to routing schemes; see, e.g., [23, 26]. A survey on labeling schemes can be found in [12].

1.5. Outline. In section 2 we give some preliminaries, and in sections 3, 4, and 5 we present the upper bounds on relationship, bi- and triconnectivity, connectivity, and sibling labeling schemes. Lower bounds for these schemes are shown in section 6 together with lower bounds for ancestor labeling schemes and the above-mentioned lower bound technique.

2. Preliminaries. For a graph G we denote the set of nodes and edges by $V(G)$ and $E(G)$. Let T be a rooted tree with n nodes. The degree of a node $v \in V(T)$, $\deg(v)$, is the number of children of v and the degree of T, $\deg(T)$, is given by $\deg(T) = \max_{v \in V(T)} \deg(v)$. Note that an edge $(v, \text{parent}(v))$ does not contribute to $\deg(v)$. The distance between two nodes $v, w \in V(T)$, denoted by $\text{dist}(v, w)$, is the number of edges on the unique simple path between v and w. The depth of v is the distance between v and the root of T. We let $T(v)$ denote the subtree of T rooted at a node $v \in V(T)$. If $w \in V(T(v))$, then v is an ancestor of w, and if $w \in V(T(v)) \backslash \{v\}$, then v is a proper ancestor of w. If v is (proper) ancestor of w, then w is a (proper) descendant of v. A node z is a common ancestor of v and w if it is an ancestor of v and w. The nearest common ancestor of v and w, $\text{nca}(v, w)$, is the common ancestor of v and w of largest depth. For a node v of depth d and $i \leq d$, the ith *level ancestor* of v, $A(v, i)$, is the ancestor of v of depth $d - i$. We call the nodes $A(v, 1)$ and $A(v, 2)$ the parent (denoted $\text{parent}(v)$) and grandparent of v, respectively. Two nodes are siblings if they have the same parent. A node with no children is a leaf and otherwise is an internal node. Two nodes in a forest are connected if and only if there is a path between them. A bit string of length n is a sequence $a = a_0 a_1 \dots a_{n-1}$, where $a_i \in \{0, 1\}$, $0 \leq i \leq n - 1$. For $0 \leq j \leq n-1$ the sequences $a_0, \dots, a_{j-1}$ and $a_{n-j}, \dots, a_{n-1}$ are the j *most significant bits* and the j *least significant bits*, respectively. The *standard binary representation* of a positive integer k is the unique bit string $a_0 \dots a_{r-1}$, where $r = \lceil \log k \rceil$ and $k = \sum_{j=0}^{r-1} a_j 2^{r-j-1}$. The *discrete logarithm* of k is the number $\lfloor \log k \rfloor$. For two integers i and j, where $i \leq j$, let $[i, j]$ be the interval $\{i, \dots, j\}$.

2.1. Labeling schemes. A *binary query* (or simply query) is a mapping $f : V(G) \times V(G) \to X$ for some set X. A *labeling scheme* for a family of graphs $\mathcal{F}$ supporting queries $f_1, \dots, f_m$ ($f_i : V(G) \times V(G) \to X_i$) is a tuple $(e, d_1, \dots, d_m)$ of mappings, where e is called the *encoder* and d_i is called the decoder for the ith query. The encoder e defines a *label assignment*, e_G, for all $G \in \mathcal{F}$, which is a mapping of $V(G)$ into bit strings called *labels*. Given the labels of two nodes v and w, the ith decoder, d_i, computes the ith query, i.e., $d_i(e_G(v), e_G(w)) = f_i(v, w)$. If the label assignment e_G is an injective mapping for all $G \in \mathcal{F}$, we say that the labeling scheme assigns *unique* labels to the nodes. A labeling scheme has *label length* bounded by s if the maximum length of the labels assigned to a node in any $G \in \mathcal{F}$ is bounded by s. We say that a labeling scheme can be computed in time t if there is an encoder e

such that for any $G \in \mathcal{F}$, e assigns labels to all nodes in G in time t.

3. Upper bound for relationship labeling schemes.

3.1. A 1-relationship labeling scheme. In this section we give a 1-relationship labeling scheme, which will serve as a basis for our k-relationship labeling scheme in the next section. As a consequence, some of the lemmas shown below will be more general than required for a 1-relationship labeling scheme. Our labeling scheme assigns unique labels to each node and supports both parent and sibling queries. As described, a labeling scheme with these properties implies a 1-relationship labeling scheme. The labeling scheme has label length bounded by $\log n + O(\log\log n)$ for trees with n nodes.

Some of the ideas in this section are inspired by [4]. There a simple labeling scheme supporting parent (but not sibling) queries is given with labels of length bounded by $\log n+O(\log\log n)$. Subsequently, they use this result to construct a more complicated labeling scheme with labels of length bounded by $\log n + O(\log^* n)$. In this section we instead generalize the simple labeling scheme supporting parent queries to also handle sibling queries within the same bounds. As noted in the introduction we later show that our labels are the smallest possible within a factor of $\log\log n$.

Let $\mathcal{T}_n$ denote the family of rooted trees with n nodes. Let $T \in \mathcal{T}_n$. As in [14] we partition T into disjoint paths. For a node $v \in V(T)$ let $\text{size}(v) = |V(T(v))|$. We classify each node of T as either *heavy* or *light* as follows. The root is light. For each internal node v we pick a child w of v of maximum size among the children of v and classify w as heavy. The remaining children are light. We call an edge to a light child a *light edge* and an edge to a heavy child a *heavy edge*. For an internal node v, let $\text{heavy}(v)$ denote the heavy child of v. Define the *light subtree*, $L(w)$, rooted at the node w as follows. If w is an internal node, then $L(w)$ is the subtree obtained from $T(w)$ by cutting away $T(\text{heavy}(w))$, and if w is a leaf $L(w) = T(w)$. Let $\text{lightsize}(v) = |V(L(v))|$. The *light depth* of a node v, $\text{lightdepth}(v)$, is the number of light edges on the path from v to the root.

LEMMA 1 (see [14]). *For any tree T with n nodes* $\text{lightdepth}(v) \leq \log n + O(1)$ *for any $v \in T$.*

The nearest light ancestor of v (possibly v itself) is denoted $\text{apex}(v)$. By removing the light edges T is partitioned into *heavy paths*.

A key ingredient of the scheme is *preorder numbers*. Order the tree T such that the rightmost child of each internal node is the heavy node. The light children need not be in any particular order. A *preorder depth first traversal* of T is obtained by first visiting the root and then recursively visiting the children of the root from left to right. The *preorder number*, $\text{pre}(v)$, is the number of nodes visited before v in this traversal, i.e., the root will have number 0 and the rightmost leaf will have number $n-1$. The labels assigned by our labeling scheme will encode $\text{pre}(v)$ in the label of v using $\lceil \log n \rceil$ bits. This will ensure that the labels are unique. In the rest of the label we will encode various smaller fields using no more than $O(\log\log n)$ bits in total. In the following we show how to test, for two nodes v and w, if one is the parent of the other or if they are siblings based on whether v and w are light or heavy nodes.

First define a node w to be a *significant ancestor* of v if $v \in L(w)$. Note that a node is its own significant ancestor. We have the following relation between significant ancestors and the preorder numbering.

LEMMA 2. *For all nodes v and w, $v \in L(w)$ if and only if* $\text{pre}(v) \in [\text{pre}(w), \text{pre}(w) + \text{lightsize}(w) - 1]$.

Proof. If w is a leaf, then $v = w$ and $\mathrm{lightsize}(w) = 1$. Hence, $\mathrm{pre}(w) = \mathrm{pre}(v) = \mathrm{pre}(w)+\mathrm{lightsize}(w)-1$ and the result follows. So assume w is an internal node. Then, in a preorder traversal, v is visited at the time of w or after and before $\mathrm{heavy}(w)$ if and only if $\mathrm{pre}(w) \leq \mathrm{pre}(v) < \mathrm{pre}(\mathrm{heavy}(w))$. Since $\mathrm{pre}(\mathrm{heavy}(w)) = \mathrm{pre}(w)+\mathrm{lightsize}(w)$ the result follows. $\square$

Consider the binary representation of $\mathrm{pre}(v)$ for an internal node v. Let $f(v) = \lfloor \log \mathrm{lightsize}(v) \rfloor$. We define the *significant preorder number*, $\mathrm{spre}(v)$, as the smallest number greater than or equal $\mathrm{pre}(v)$ which is a multiple of $2^{f(v)}$. Equivalently,

$$\mathrm{spre}(v) = \begin{cases} \mathrm{pre}(v) & \text{if } \mathrm{pre}(v) \bmod 2^{f(v)} = 0, \\ \mathrm{pre}(v) - (\mathrm{pre}(v) \bmod 2^{f(v)}) + 2^{f(v)} & \text{otherwise.} \end{cases}$$

The following lemma states the relations we need between the preorder and significant preorder numbers.

LEMMA 3. *For all nodes v and w the following hold:*

(i) $\mathrm{spre}(v) \in [\mathrm{pre}(v), \mathrm{pre}(v) + \mathrm{lightsize}(v) - 1]$.

(ii) *$v = w$ if and only if* $\mathrm{lightdepth}(v) = \mathrm{lightdepth}(w)$ *and* $\mathrm{spre}(v) = \mathrm{spre}(w)$.

(iii) *If* $\mathrm{lightdepth}(v) = \mathrm{lightdepth}(w)$, *then* $\mathrm{pre}(w) < \mathrm{pre}(v)$ *if and only if* $\mathrm{spre}(w) < \mathrm{spre}(v)$.

Proof. (i) If $\mathrm{pre}(v) \bmod 2^{f(v)} = 0$, then $\mathrm{spre}(v) = \mathrm{pre}(v)$, and since $\mathrm{lightsize}(v) \geq 1$ for all v the result follows. Otherwise $1 \leq \mathrm{pre}(v) \bmod 2^{f(v)} \leq 2^{f(v)} - 1$. Hence, $\mathrm{spre}(v) \geq \mathrm{pre}(v) - (2^{f(v)} - 1) + 2^{f(v)} = \mathrm{pre}(v) + 1$ and $\mathrm{spre}(v) \leq \mathrm{pre}(v) - 1 + 2^{f(v)} \leq \mathrm{pre}(v) - 1 + \mathrm{lightsize}(v)$.

(ii) If $v = w$, the conditions are clearly satisfied. Conversely, assume that $v \neq w$ and the conditions are satisfied. Since $v \neq w$ and $\mathrm{lightdepth}(v) = \mathrm{lightdepth}(w)$ we have that $v \notin L(w)$ and $w \notin L(v)$. Then, by Lemma 2 $\mathrm{pre}(v) \notin [\mathrm{pre}(w), \mathrm{pre}(w) + \mathrm{lightsize}(w) - 1]$ and $\mathrm{pre}(w) \notin [\mathrm{pre}(v), \mathrm{pre}(v)+\mathrm{lightsize}(v)-1]$, and hence these intervals must be disjoint. However, since $\mathrm{spre}(v) = \mathrm{spre}(w)$ we have, by (i), the contradiction that $\mathrm{spre}(v) \in [\mathrm{pre}(v), \mathrm{pre}(v) + \mathrm{lightsize}(v) - 1]$ and $\mathrm{spre}(v) \in [\mathrm{pre}(w), \mathrm{pre}(w) + \mathrm{lightsize}(w) - 1]$.

(iii) Assume that $\mathrm{lightdepth}(v) = \mathrm{lightdepth}(w)$. If $\mathrm{pre}(w) < \mathrm{pre}(v)$, then $v \notin L(w)$. By Lemma 2, $\mathrm{pre}(v) \notin [\mathrm{pre}(w), \mathrm{pre}(w) + \mathrm{lightsize}(w) - 1]$ and since $\mathrm{pre}(w) < \mathrm{pre}(v)$ we have $\mathrm{pre}(w) + \mathrm{lightsize}(w) - 1 < \mathrm{pre}(v)$. By (i) it follows that $\mathrm{spre}(w) < \mathrm{spre}(v)$. Conversely, since $\mathrm{spre}(w) < \mathrm{spre}(v)$ and $\mathrm{lightdepth}(v) = \mathrm{lightdepth}(w)$ we have by (ii) that $v \neq w$. Furthermore, as in the proof of (ii), this implies that the intervals $[\mathrm{pre}(v), \mathrm{pre}(v) + \mathrm{lightsize}(v) - 1]$ and $[\mathrm{pre}(w), \mathrm{pre}(w) + \mathrm{lightsize}(w) - 1]$ are disjoint. By (i), $\mathrm{spre}(v) \in [\mathrm{pre}(v), \mathrm{pre}(v) + \mathrm{lightsize}(v) - 1]$ and $\mathrm{spre}(w) \in [\mathrm{pre}(w), \mathrm{pre}(w)+\mathrm{lightsize}(w)-1]$ and since these intervals are disjoint and $\mathrm{spre}(w) < \mathrm{spre}(v)$ we have that $\mathrm{pre}(w) < \mathrm{pre}(v)$. $\square$

Note that by Lemma 3(ii) it follows that any node v is uniquely identified by $\mathrm{spre}(v)$ and $\mathrm{lightdepth}(v)$. The following lemma shows that the significant preorder number of a significant ancestor can be represented efficiently. In particular, $\mathrm{spre}(\mathrm{parent}(v))$ can be represented efficiently if v is a light node.

LEMMA 4. *Given* $\mathrm{pre}(v)$ *we can represent* $\mathrm{spre}(w)$ *for each significant ancestor w of v using only* $\log\log n + O(1)$ *bits per significant ancestor.*

Proof. Let w be a significant ancestor of v. Since $\mathrm{lightsize}(w) < 2^{f(w)+1}$ there can be, apart from $\mathrm{spre}(w)$, at most one other number in the interval $[\mathrm{pre}(w), \mathrm{pre}(w) + \mathrm{lightsize}(w) - 1]$ with all the $f(w)$ least significant bits set to zero, i.e., the number $\mathrm{spre}(w) + 2^{f(w)}$. Let $\mathrm{pre}'(v)$ be $\mathrm{pre}(v)$ with all the $f(w)$ least significant bits set to zero. Since w is a significant ancestor of v, $v \in L(w)$ and thus, by Lemma 2, $\mathrm{pre}(v) \in$

$[\mathrm{pre}(w), \mathrm{pre}(w) + \mathrm{lightsize}(w) - 1]$. Hence, $\mathrm{pre}'(v)$ will be either $\mathrm{spre}(w) - 2^{f(w)}$, $\mathrm{spre}(w)$ or $\mathrm{spre}(w) + 2^{f(w)}$ and therefore $\mathrm{spre}(w)$ is either $\mathrm{pre}'(v) + 2^{f(w)}$, $\mathrm{pre}'(v)$, or $\mathrm{pre}'(v) - 2^{f(w)}$. Clearly, representing $f(w)$ and two extra bits to distinguish these three cases we can compute $\mathrm{spre}(w)$ from $\mathrm{pre}(v)$. This can be represented by $\lceil \log\log n \rceil + 2$ bits since $f(w)$ is bounded by $\log n$. □

For each light node v we will encode $\mathrm{lightdepth}(v)$, $\mathrm{spre}(v)$, and $\mathrm{spre}(\mathrm{parent}(v))$ in the label of v. By Lemma 1 $\mathrm{lightdepth}(v) \leq \log n + O(1)$ and can thus be represented using $\log\log n + O(1)$ bits. Since the labels encode $\mathrm{pre}(v)$ and v is light, we have by Lemma 4 that $\mathrm{spre}(v)$ and $\mathrm{spre}(\mathrm{parent}(v))$ can also be represented using $\log\log n + O(1)$ bits. By Lemma 3(ii), $\mathrm{lightdepth}(v)$ together with $\mathrm{spre}(v)$ uniquely identifies the node v. This immediately implies the following.

LEMMA 5. *For a light node v and internal node w, w is the parent of v if and only if* $\mathrm{lightdepth}(v) = \mathrm{lightdepth}(w) + 1$ *and* $\mathrm{spre}(\mathrm{parent}(v)) = \mathrm{spre}(w)$.

LEMMA 6. *For two light nodes v and w, w and v are siblings if and only if* $\mathrm{lightdepth}(v) = \mathrm{lightdepth}(w)$ *and* $\mathrm{spre}(\mathrm{parent}(v)) = \mathrm{spre}(\mathrm{parent}(w))$.

Next we show how to handle the remaining cases. Define $\mathrm{diff\_parent}(v) = \mathrm{spre}(v) - \mathrm{spre}(\mathrm{parent}(v))$ and leave it undefined for the root. Similarly, for internal nodes, define $\mathrm{diff\_heavy}(v) = \mathrm{spre}(\mathrm{heavy}(v)) - \mathrm{spre}(v)$ and leave it undefined for leaves. The following lemma shows how the discrete logarithm of $\mathrm{diff\_parent}(v)$ and $\mathrm{diff\_heavy}(v)$ can be used to test for parenthood between two nodes on a heavy path. Since the discrete logarithm is bounded by $\log n$, only $\lceil \log\log n \rceil$ bits are needed to represent each of these numbers.

LEMMA 7. *For heavy node v and internal node w, w is the parent of v if and only if* $\mathrm{spre}(w) < \mathrm{spre}(v)$, $\mathrm{lightdepth}(v) = \mathrm{lightdepth}(w)$ *and* $\lfloor \log(\mathrm{spre}(v) - \mathrm{spre}(w)) \rfloor = \lfloor \log \mathrm{diff\_parent}(v) \rfloor = \lfloor \log \mathrm{diff\_heavy}(w) \rfloor$

Proof. For $w = \mathrm{parent}(v)$ it is straightforward, using Lemma 3, to verify that the conditions are satisfied. Conversely, assume that a node $w \neq \mathrm{parent}(v)$ satisfies the conditions. Since $\mathrm{spre}(w) < \mathrm{spre}(v)$ and $\mathrm{lightdepth}(v) = \mathrm{lightdepth}(w)$, we have by Lemma 3(iii) that $\mathrm{pre}(w) < \mathrm{pre}(v)$, and therefore w cannot be a descendant of v. Furthermore, w cannot be a descendant of any other sibling of v, because then $\mathrm{lightdepth}(w) > \mathrm{lightdepth}(v)$. It follows that w cannot be a descendant of $\mathrm{parent}(v)$. Hence, in a preorder traversal of T the node $\mathrm{heavy}(w)$ is visited before $\mathrm{parent}(v)$ or $\mathrm{heavy}(w) = \mathrm{parent}(v)$. That is, $\mathrm{pre}(\mathrm{heavy}(w)) \leq \mathrm{pre}(\mathrm{parent}(v))$ and by Lemma 3(ii) and (iii), also $\mathrm{spre}(\mathrm{heavy}(w)) \leq \mathrm{spre}(\mathrm{parent}(v))$, and therefore $\mathrm{spre}(v) - \mathrm{spre}(w) \geq (\mathrm{spre}(\mathrm{heavy}(w)) - \mathrm{spre}(w)) + (\mathrm{spre}(v) - \mathrm{spre}(\mathrm{parent}(v))) = \mathrm{diff\_heavy}(w) + \mathrm{diff\_parent}(v)$. By the identities $\lfloor \log(\mathrm{spre}(v) - \mathrm{spre}(w)) \rfloor = \lfloor \log \mathrm{diff\_parent}(v) \rfloor = \lfloor \log \mathrm{diff\_heavy}(w) \rfloor$ this leads to the contradiction $\mathrm{spre}(v) - \mathrm{spre}(w) \geq \mathrm{diff\_heavy}(w) + \mathrm{diff\_parent}(v) \geq 2 \cdot 2^{\lfloor \log \mathrm{diff\_parent}(v) \rfloor} = 2 \cdot 2^{\lfloor \log(\mathrm{spre}(v) - \mathrm{spre}(w)) \rfloor} > \mathrm{spre}(v) - \mathrm{spre}(w)$. □

Considering siblings instead, we immediately obtain the following corollary to Lemma 7.

LEMMA 8. *A heavy node v and light node w are siblings if and only if* $\mathrm{spre}(\mathrm{parent}(w)) < \mathrm{spre}(v)$, $\mathrm{lightdepth}(v) = \mathrm{lightdepth}(w) - 1$, *and* $\lfloor \log(\mathrm{spre}(v) - \mathrm{spre}(\mathrm{parent}(w))) \rfloor = \lfloor \log \mathrm{diff\_parent}(v) \rfloor = \lfloor \log \mathrm{diff\_heavy}(\mathrm{parent}(w)) \rfloor$.

Note that since any node has at most one heavy child, two heavy nodes v and w are siblings if and only if $v = w$. Since the labels are unique it is trivial to handle this case.

Combining the above lemmas we obtain the 1-relationship labeling scheme. For $T \in \mathcal{T}_n$ let the encoder $e_T(v), v \in V(T)$, encode $\mathrm{pre}(v)$, $\mathrm{lightdepth}(v)$, $\mathrm{spre}(v)$,

$\lfloor \log \text{diff\_heavy}(v) \rfloor$ and a type bit indicating if v is a light or heavy node. Furthermore, if v is a light node encode $\text{spre}(\text{parent}(v))$ and $\lfloor \log \text{diff\_heavy}(\text{parent}(v)) \rfloor$. If v is a heavy node encode $\lfloor \log \text{diff\_parent}(v) \rfloor$. As described, $\text{pre}(v)$ uses $\lceil \log n \rceil$ bits and each of the other values uses $\log \log n + O(1)$ bits each. For easy decoding we represent each of the values in fixed sized fields in the label of v. The first $\lceil \log n \rceil$ bits stores $\text{pre}(v)$. The other values are represented, in five fields (we leave one field undefined when v is a light node) of the same length, in the next $5 \log \log n + O(1)$ bits. At the end of the label we store the type bit. We will assume that the decoder does not know the value n, i.e., the decoder is not specialized to trees of size n but will work with any tree, regardless of its size. Due to this restriction we cannot compute $\lceil \log n \rceil$ directly and use this to extract the preorder number and then the rest of the fields. Instead we use a self-delimiting code for $\lceil \log n \rceil$. In particular, we prefix the label with $1^{|x|}0x$, where x is the binary representation of the length of the field containing $\text{pre}(v)$. Since the length of $\text{pre}(v)$ is $\lceil \log n \rceil$, we have added only $2 \log \log n + O(1)$ bits. Note that the unary prefix $1^{|x|}0$ enables us to figure out the length of x. In total the label length will be bounded by $\log n + O(\log \log n)$. By uniqueness of the labels and Lemmas 5 through 8, it is straightforward to construct decoders testing if two nodes are $(0,0)$-, $(0,1)$-, $(1,0)$-, or $(1,1)$-related. In summary we have the next theorem.

Theorem 1. *For trees with n nodes there is a 1-relationship labeling scheme with label length bounded by $\log n + O(\log \log n)$.*

Finally, note that labels for all nodes in T can be computed in $O(n)$ time and queries can be implemented in $O(1)$ time per query assuming standard binary operations on a RAM.

3.2. A general k-relationship labeling scheme.

In this section we generalize the result of the previous section to a k-relationship labeling scheme. The scheme extends the ideas of the first labeling scheme and has label length bounded by $\log n + O(k^2(\log \log n + \log k))$, which for constant k is $\log n + O(\log \log n)$.

We first extend the definition of $\text{diff\_heavy}(v)$ and $\text{diff\_parent}(v)$ as follows. If v has a descendant u on the same heavy path as v of distance m, let $\text{diff\_heavy}(v,m) = \text{spre}(u) - \text{spre}(v)$, and if there is no such node u let $\text{diff\_heavy}(v,m) = 2n$, i.e., the discrete logarithm of $2n$ will be $\lfloor \log n \rfloor + 1$, indicating that this is not an actual difference. Similarly, define $\text{diff\_parent}(v,m)$ for the ancestor on the same heavy path of v of distance m. Furthermore, for a node v we define the *index* of v, $\text{index}(v)$, as the number of nodes with the same light depth as v and with smaller preorder numbers than v. We will use the following generalization of Lemma 7.

Lemma 9. *For a heavy node v and internal node w, w and v are on the same heavy path and w is an ancestor of v of distance $m \geq 1$ if and only if* $\text{spre}(w) < \text{spre}(v)$, $\text{lightdepth}(v) = \text{lightdepth}(w)$, $\lfloor \log(\text{spre}(v) - \text{spre}(w)) \rfloor = \lfloor \log \text{diff\_parent}(v,m) \rfloor = \lfloor \log \text{diff\_heavy}(w,m) \rfloor$, *and* $\text{index}(v) \bmod m = \text{index}(w) \bmod m$.

Proof. Let x denote the ancestor of v of distance m on the heavy path of v. Similarly, let y denote the descendant of w of distance m on the heavy path of w. If x or y does not exist, then the conditions do not hold by definition of diff_parent and diff_heavy. If they both exist and if $x = w$ (or equivalently $y = v$), it is straightforward to check that the conditions are satisfied. Conversely, assume that the conditions are satisfied and $x \neq w$. Since $\lfloor \log(\text{spre}(v) - \text{spre}(w)) \rfloor = \lfloor \log \text{diff\_parent}(v,m) \rfloor = \lfloor \log \text{diff\_heavy}(w,m) \rfloor$, both x and y exist and are on the same heavy paths as v and w, respectively. As in the proof of Lemma 7, we have $\text{lightdepth}(w) = \text{lightdepth}(y) = \text{lightdepth}(x) = \text{lightdepth}(v)$ and $\text{pre}(w) < \text{pre}(v)$.

Since index(w) mod m = index(y) mod m = index(x) mod m = index(v) mod m and $x \neq w$, the paths from w to y and x to v are either disjoint or $x = y$. Thus pre(y) $\leq$ pre(x) and by Lemma 3(ii) and (iii) also spre(y) $\leq$ spre(x). Therefore spre(v) $-$ spre(w) $\geq$ diff_heavy(w, m) + diff_parent(v, m). By the identities $\lfloor \log(\mathrm{spre}(v) - \mathrm{spre}(w)) \rfloor = \lfloor \log \mathrm{diff\_parent}(v,m) \rfloor = \lfloor \log \mathrm{diff\_heavy}(w,m) \rfloor$, we obtain the contradiction $\mathrm{spre}(v) - \mathrm{spre}(w) \geq \mathrm{diff\_heavy}(w,m) + \mathrm{diff\_parent}(v,m) \geq 2 \cdot 2^{\lfloor \log \mathrm{diff\_parent}(v,m) \rfloor} = 2 \cdot 2^{\lfloor \log(\mathrm{spre}(v) - \mathrm{spre}(w)) \rfloor} > \mathrm{spre}(v) - \mathrm{spre}(w)$. ◻

The main idea in our labeling scheme is to store, in the label of v, pre(v) and lightdepth(v) as before. Furthermore, for each significant ancestor w of v of distance at most k we will represent spre(w) together with diff_heavy(w, m), diff_parent(w, m), and index(w) mod m for $1 \leq m \leq k$. Then, to test if two nodes v and w are (k_1, k_2)-related we identify the heavy path containing the nearest common ancestor of v and w and compute distances to and on this heavy path.

3.3. The encoder. We can now describe the encoder for our k-relationship labeling scheme. For $T \in \mathcal{T}_n$ let the label $e_T(v)$, $v \in V(T)$ encode pre(v) and lightdepth(v). Furthermore, we store an *ancestor table* of s entries, where s is the number of significant ancestors of distance at most k from v. If w is the ith significant ancestor of v, the ith entry in the ancestor table will represent spre(w), dist(v, w), and a single bit, called the *apex bit*, indicating whether the distance dist(w, apex(w)) is at most k. If this is so we store dist(w, apex(w)) and otherwise leave this field undefined. Furthermore, the ith entry also represents, for $1 \leq m \leq k$, diff_heavy(w, m), diff_parent(w, m) and index(w) mod m. Hence, number of bits used to represent an entry is bounded by $O(k \log\log n + k \log k)$ and thus the total number of bits used for the ancestor table is at most $O(k^2(\log\log n + \log k))$. Note that since w is the ith significant ancestor we have that lightdepth(w) = lightdepth(v) $- i$ and hence this information is implictly stored in the table.

For efficient computation of the queries we store a *lookup table* of k entries. The ith entry stores the light depth of $A(v, i)$. Hence the lookup table uses at most $O(k \log\log n)$ bits. As before all the values are stored in fixed sized fields and we prefix the label with small codes representing the length of pre(v) and each of tables. In total the label length is bounded by $\log n + O(k^2(\log\log n + \log k))$. Computing the tables can be done in $O(k)$ time per node after $O(n)$ time preprocessing and hence the labeling scheme can be computed in $O(nk)$ time.

3.4. The decoder. In the following we present the decoder for our k-relationship labeling scheme. We first present necessary and sufficient conditions for two nodes v and w to be (k_1, k_2)-related and then show how to test these conditions using only the labels of v and w.

LEMMA 10. *Let $v, w \in T$ and distances k_1 and k_2 (not both zero) be given. Let v' be the significant ancestor of v such that* lightdepth(v') = lightdepth($A(v, k_1)$) *and if $v' \neq v$ let v'' be the significant ancestor of v of light depth* lightdepth($A(v, k_1)$) + 1. *Otherwise let $v'' = v$. Similarly, define w' and w'' for w and k_2. Then, v and w are (k_1, k_2)-related if and only if one of the following disjoint conditions is satisfied:*

(i) *$v' = w'$, v'', and w'' are on different heavy paths,* dist(v, v') = k_1 *and* dist(w, w') = k_2.

(ii) *v' and w' are on same heavy path, v' is a proper ancestor of w',* dist(w', v') = $k_2 -$ dist(w, w'), *and* dist(v, v') = k_1.

(iii) *v' and w' are on same heavy path, w' is a proper ancestor of v',* dist(v', w') = $k_1 -$ dist(v, v'), *and* dist(w, w') = k_2.

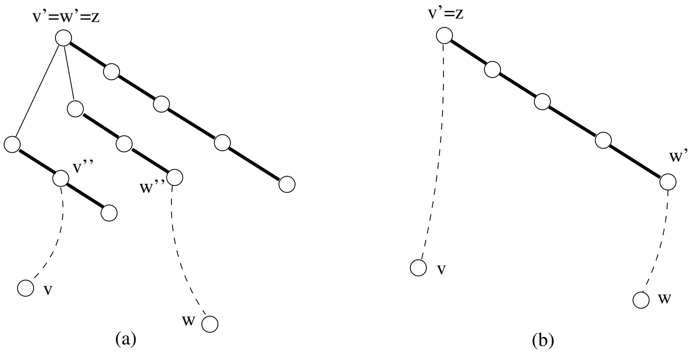


Fig. 1. *Cases for Lemma* 10: (a) *case* (i), (b) *case* (ii).

Proof. The situation is illustrated in Figure 1. Let $z = \text{nca}(v, w)$. If one of the conditions is satisfied it is straightforward to check that v and w are (k_1, k_2)-related. Conversely, if v and w are (k_1, k_2)-related, then z must be on the heavy path of v' and w' and $z = v'$ or $z = w'$. If $z = v' = w'$, then z is a significant ancestor of both v and w. Hence, since k_1 and k_2 are not both zero, v'' and w'' must be on different heavy paths; otherwise there would be a common ancestor of larger depth than z contradicting the assumption that $z = \text{nca}(v, w)$. If $z = v' \neq w'$, then v' is a proper ancestor of w', and if $z = w' \neq v'$, then w' is a proper ancestor of v'. Since v and w are (k_1, k_2)-related the distance conditions are satisfied. □

Given only the labels of the nodes v and w we can test if they are (k_1, k_2)-related for $k_1, k_2 \leq k$ as follows. First, since the labels are unique, it is trivial to test if v and w are $(0, 0)$-related. Hence, we will assume that k_1 and k_2 are not both zero. We will show how to test each of the conditions in Lemma 10 using only the labels. Using the lookup tables we first compute the entries in the ancestor tables for the nodes v', v'', w', and w''. Assume that the values stored at these entries of the tables are available. Using Lemma 3(ii) we can check if $v' = w'$. The distances $\text{dist}(v, v')$ and $\text{dist}(w, w')$ are stored directly in the ancestor tables of v and w, and the first three conditions in (ii) and (iii) can be checked using Lemma 9. What remains is to describe how to test if v'' and w'' are on different heavy paths. Since the distances $\text{dist}(v'', \text{apex}(v''))$ and $\text{dist}(w'', \text{apex}(w''))$ are both smaller than k, they are available in the ancestor tables. If v'' and w'' are on the same heavy path their distance must be $|\text{dist}(v'', \text{apex}(v'')) - \text{dist}(w'', \text{apex}(w''))|$, so we can use Lemma 9 to test whether they are on the same heavy path and if so if they are this distance apart. Thus we have shown that the conditions of Lemma 10 can be tested using only the labels of v and w and so we can determine if v and w are (k_1, k_2)-related. In summary we have shown the next theorem.

Theorem 2. *For trees with n nodes there is a k-relationship labeling scheme with label length bounded by* $\log n + O(k^2(\log\log n + \log k))$.

As noted, the k-relationship labeling scheme can be computed in $O(nk)$ time and due to the lookup and ancestor tables queries can be performed in $O(1)$ time.

4. Upper bounds for bi- and triconnectivity labeling schemes.

As an application of our k-relationship labeling scheme of section 3 we give a labeling scheme for biconnectivity. Subsequently, we use a reduction from [21] to obtain a labeling scheme for triconnectivity. Both labeling schemes assign unique labels. For a graph G with n nodes, the labeling scheme for bi- and triconnectivity uses labels of length bounded by $\log n + O(\log \log n)$ and $3 \log n + O(\log \log n)$, respectively.

We first give some preliminaries. Let G be a graph. A set of paths P connecting two nodes v and w in G is *vertex-disjoint* if each node except v and w appears in at most one path $p \in P$. We define v and w to be *m-vertex connected* if there is a set of vertex-disjoint paths of size m connecting v and w. We say that v and w are bi- or triconnected if they are 2- or 3-vertex connected, respectively. A *cut-node* is a node whose removal (and all incident edges) disconnects the graph. A *block* of a graph G is a maximal connected subgraph without a cut-node. By maximality, different blocks of G overlap in at most one node, which is then the cut-node. Using Menger's theorem (see, e.g., [9]), it can be shown that two nodes $v, w \in V(G)$ are biconnected if and only if they are within the same block and the block has at least three nodes.

We define the *block graph* B of G. Each node in G is represented by a unique node in B and each node in B either represents a node in G or a block with at least three nodes in G. The edges of B are defined as follows. Let v be a node in G and let $B(v)$ denote the set of blocks in G that contain v and have at least three nodes. For each node representing a block $b \in B(v)$ there is an edge to the node representing v in B. A node in B representing a node in G that is not contained in any block with at least three nodes is not incident to any other node in B. By the maximality of blocks we have the next lemma.

LEMMA 11. *The block graph B of a graph G is a forest of unrooted trees.*

Using depth-first search [24], we can compute the block forest in linear time. We root each tree in the forest as follows. If the tree contains only one node, this node is the root. Otherwise the tree contains at least one node representing a block and we arbitrarily root the tree in such a node. By B_r we denote the rooted version of the block forest B.

LEMMA 12. *Two nodes v and w in G are biconnected in G if and only if, in the block forest of rooted trees B_r, either v and w are siblings, v is the grandparent of w, or w is the grandparent of v.*

Proof. If v and w are biconnected in G, then they are contained in the same block with at least three nodes, and hence they are incident to the same node representing a block. In B_r, this implies that v and w are either siblings or one is the grandparent of the other. Conversely, if v and w are siblings or one is the grandparent of the other in B_r, then they are incident to the same node representing a block. Hence, they are contained in the same block with at least three nodes and are thus biconnected. □

To test the conditions in Lemma 12 we extend our k-relationship labeling scheme to handle the more general case of forests. Add an extra root node connected to each root of the trees in the forest. This produces a tree where we then apply our k-relationship labeling scheme. The modifications needed to handle a special root node are straightforward to implement. Using a 2-relationship labeling scheme for the forest B_r we obtain by Lemma 12 the following theorem.

THEOREM 3. *For graphs with n nodes there is a biconnectivity labeling scheme that assigns unique labels with label length bounded by $\log n + O(\log \log n)$.*

Since we can compute the block forest B_r in $O(n)$ time, the labeling scheme can be computed in $O(n)$ time and with the 2-relationship labeling scheme queries can be answered in $O(1)$ time.

As noted in the introduction we can use our biconnectivity labeling scheme to obtain a triconnectivity labeling scheme using a reduction from [21]. There a labeling scheme for triconnectivity is given using labels of length bounded by $5 \log n$. By Lemmas 3.3, 3.4, and 3.6 in [21] and Theorem 3 we obtain the following improvement.

Theorem 4. *For graphs with n nodes there is a triconnectivity labeling scheme that assigns unique labels with label length bounded by* $3\log n + O(\log\log n)$.

5. Upper bounds for sibling and connectivity labeling schemes. In this section we consider labeling schemes for sibling queries and connectivity queries in a forest. First we consider sibling queries. If two nodes in the same tree can be given the same label, we can label the nodes with labels of length $\lceil \log n \rceil$ as follows. Partition the nodes into groups such that two nodes are siblings if and only if they belong to the same group. This construction gives $g \leq n$ groups, which are numbered $1, 2, \ldots, g$. Nodes in the same group are given the same label, namely, the number of the group. Now, two nodes are siblings if and only if they have the same label.

Theorem 5. *For trees with n nodes there is a sibling labeling scheme with label length bounded by* $\lceil \log n \rceil$.

Next we show how to assign unique labels for trees with maximum degree Δ. We group the nodes as above. We assign to each node v two numbers: a group number $g(v)$ to answer sibling queries as above, and an individual number $i(v)$ to make its label unique. Two nodes in the same group will be given the same group number. Assume we have g groups $g_1, g_2, \ldots, g_g$. Let $|g_i|$ be the number of nodes in g_i. Using a Huffman code [15] we give each node in group g_i, a group number of length $\log n - \log |g_i| + O(1)$. The individual numbers given to the nodes in group g_i are simply $1, 2, \ldots, |g_i|$, of length $\log |g_i| + O(1)$. In total we use $\log n + O(1)$ bits for the group and individual numbers; however, coding these two numbers as one label, we also need to be able to separate these two numbers given the label of a node. We use the first $O(\log\log\Delta)$ bits of the label to code the length of the individual number as follows. The individual number in a tree with maximum degree Δ is at most Δ and can be represented with at most $q = \log\Delta + O(1)$ bits. To represent the length of the individual number we need $O(\log q) = O(\log\log\Delta)$ bits. Now, we also need to represent the length of the bit string representing the length of the individual number, but this can be done simply by using an unary code of length $O(\log\log\Delta)$.

Theorem 6. *For trees with n nodes and maximum degree Δ there is a sibling labeling scheme that assigns unique labels with label length bounded by* $\log n + O(\log\log\Delta)$.

Using the same observations, grouping connected nodes, we get the next theorem.

Theorem 7. *For forests with n nodes there is a connectivity labeling scheme that assigns unique labels with label length bounded by* $\log n + O(\log\log n)$.

It is straightforward to compute the above labeling schemes in $O(n)$ time and answer queries in $O(1)$ time assuming standard binary operations on a RAM.

6. Lower bounds. In this section we present a lower bound technique and subsequently give lower bounds for ancestor, connectivity, sibling, 1-relationship, 2-restricted distance, and biconnectivity labeling schemes.

If v is an ancestor of w *or* w is an ancestor of v, we say that v and w are weak ancestors. A lower bound for a weak ancestor labeling scheme is clearly a lower bound for an ancestor labeling scheme. The lower bound presented in this paper is for weak ancestor labeling schemes.

We will use the following technique to show this lower bound. First we give a family of trees $\mathcal{F}_{\mathcal{A}}$ where each tree consists of cn nodes for a constant c. We then show that any labeling scheme (which may use nonunique labels) for weak ancestor queries needs to use $\Omega(n \log n)$ different labels for $\mathcal{F}_{\mathcal{A}}$. If m different labels are necessary, then the label length must be at least $\log m$. Since $\log(cn\log n) = \log n + \Omega(\log\log n)$, for any constant c, we establish the lower bound. A similar construction is used for the other lower bounds.

In some cases, e.g., in [7], the goal is to minimize the average length of labels instead of the maximum. We note that, using the above technique, our lower bounds also hold for the average length of labels.

6.1. Lower bound technique. Let $\mathcal{S}$ be a set of elements and let $e : \mathcal{S} \to \mathcal{D}$ be a function labeling $\mathcal{S}$ with elements from some domain $\mathcal{D}$. We will assume $|\mathcal{S}| = nk$, where k is an integer $\leq \log n$ and n is a power of two. We define a partition P of $\mathcal{S}$ into k boxes each of n elements. The elements in the ith box, $1 \leq i \leq \log n$, denoted by B_i are partitioned into $n/2^i$ *groups* each of 2^i elements.

LEMMA 13. *Let S, e, and k be as described above. If there exists a partition P such that the following two properties hold, then $|\mathcal{D}| = \Omega(nk)$:*

(i) *for two different elements $s_1, s_2 \in \mathcal{S}$, if s_1 and s_2 belong to the same box, then $e(s_1) \neq e(s_2)$,*

(ii) *for elements $s_1, s_2, s_3, s_4 \in \mathcal{S}$, if s_1 and s_2 belong to two different groups in the same box, $e(s_1) = e(s_3)$ and $e(s_2) = e(s_4)$, then s_3 and s_4 belong to two different groups.*

Proof. We will say the function e associates labels with the elements from $\mathcal{S}$. The elements associated with the same label are called *neighbors*. In the following we give a strategy to choose a subset S' of elements from $\mathcal{S}$, guaranteeing that for all $s_1, s_2 \in S'$, where $s_1 \neq s_2$, s_1 and s_2 will not be neighbors. We call a strategy with such a guarantee a *safe* strategy. The number of labels needed by e for $\mathcal{S}$ will be at least the size of S' since $|\mathcal{D}| \geq |S'|$ when S' is chosen by a safe strategy. We say an element is a *marked* element if it is chosen to belong to S'. Hence, no two elements with the same label will be marked. If one or more elements from a group are marked we say the group is marked. For a box B we let $M(B)$ denote the number of marked groups belonging to the box.

We first mark elements from the box B_k and next for B_i in order of decreasing i. All elements in B_k will be marked. From the first property of Lemma 13 there are no neighbors in the same box and the marking is therefore safe. When marking elements from the remaining boxes B_i, $i < k$, we keep the invariant that $M(B_i) \leq n/2^{i+1}$. Hence, we will mark elements from at most half of the groups belonging to B_i.

Let $F(i)$ be the set of groups belonging to the boxes B_j, $j \geq i$, and let $M(F(i))$ be the number of marked groups belonging to $F(i)$. Since we keep the invariant that $M(B_i) \leq n/2^{i+1}$ for $i < k$, we have that for $i \leq k$, $M(F(i)) \leq n/2^k + \sum_{j=i}^{k-1} n/2^{j+1} = n/2^i$. Next, we describe how to mark elements from B_i, after marking elements from B_j, $j > i$. If a group in B_i includes an element with a marked neighbor in B_j, $j > i$, we say the group is *closed*. If a group is not closed it is *open*.

Let $s_1, s_2 \in B_i$ belong to two different groups. If s_1 has a marked neighbor s_3 and s_2 has a marked neighbor s_4, then by the second property of Lemma 13, s_3 and s_4 must belong to two different marked groups from $F(i+1)$. Hence, for each closed group in B_i we can associate a marked group from $F(i+1)$ which will not be associated to any other group in B_i. Since the number of groups in B_i is $n/2^i$ and we keep the invariant that $M(F(i+1)) \leq n/2^{i+1}$, at least $n/2^{i+1}$ of the groups in B_i will be open. Since the elements from the open groups do not have a marked neighbor and none of them are neighbors by the first property of Lemma 13 it is safe to mark all elements from the $n/2^{i+1}$ open groups of B_i. This way we maintain the invariant of marking elements from at most half of the groups in B_i, $i < k$. Summarizing, we mark all elements in B_k and half the elements from the remaining $k-1$ boxes. In total we mark $\Omega(nk)$ elements. □

In the following sections we will define different families of graphs for which the nodes from these graphs can be partitioned such that the labeling obeys the properties given in Lemma 13.

6.2. Ancestor labeling schemes. To show a lower bound for a weak ancestor labeling scheme we give a family $\mathcal{F}_{\mathcal{A}}$ of $\log n$ trees $\{T_1, T_2, \dots, T_{\log n}\}$, each of size $2n+1$. We show that for a subset $\mathcal{S}$ of the nodes from $\mathcal{F}_{\mathcal{A}}$, where $|\mathcal{S}| = n \log n$, there is a partition P of $\mathcal{S}$, such that any e must obey the two properties in Lemma 13. This implies that at least $\Omega(n \log n)$ labels are needed and will conclude our proof.

The tree T_i in $\mathcal{F}_{\mathcal{A}}$ consists of a root node with $n/2^i$ children. Each child v is the root of a path $\rho(v)$ of length 2^i. Furthermore, each node on these paths has a child which is a leaf not belonging to the path.

We have $|V(T_i)| = 2(n/2^i)2^i+1 = 2n+1$. We let $\mathcal{S}$ be the subset of nodes from $\mathcal{F}_{\mathcal{A}}$ which belongs to a path $\rho(v)$, where v is a child of one of the root nodes in the family. Hence, $|\mathcal{S}| = n \log n$. Box B_i is the subset of nodes from $\mathcal{S}$ which belongs to the tree T_i. The nodes from box B_i are partitioned into groups such that two nodes from the same group belong to the same path. Next we show that the two properties from Lemma 13 must be fulfilled for any weak ancestor labeling scheme (e, d) in this partition.

Consider the first property. Let $s_1, s_2 \in B_i$, $s_1 \neq s_2$. If s_1 and s_2 are weak ancestors, choose s_2 to be the node closer to the root. On the other hand, if s_1 and s_2 are not weak ancestors, then choose s_2 arbitrarily. Let c be the leaf in T_i which is the child of s_2. Note that in both cases s_1 and c are not weak ancestors and therefore $d(e(s_1), e(c)) \neq d(e(s_2), e(c))$, which implies that $e(s_1) \neq e(s_2)$.

Next we consider the second property. Let $s_1, s_2, s_3, s_4 \in \mathcal{S}$, where s_1 and s_2 belong to two different groups in the same box. This implies that s_1 and s_2 are not weak ancestors. Hence, if $e(s_1) = e(s_3)$ and $e(s_2) = e(s_4)$, then s_3 and s_4 are not weak ancestors and therefore s_3 and s_4 must belong to different groups.

THEOREM 8. *A weak ancestor labeling scheme for trees with n nodes needs label of length* $\log n + \Omega(\log \log n)$.

6.3. Connectivity labeling schemes. In this section we consider the minimum label length required to answer connectivity queries in a forest if the labels assigned to the nodes must be unique. Let $\mathcal{F}_{\mathcal{C}}$ be the family of $\log n$ forests F_i, $1 \leq i \leq \log n$, where F_i consist of $2^{\log n - i}$ paths of length 2^i. We have $|V(F_i)| = n$. We let $\mathcal{S}$ be the nodes from $\mathcal{F}_{\mathcal{C}}$. Box B_i is the nodes from F_i. The nodes in B_i are partitioned into groups such that connected nodes are in the same group.

The first property from Lemma 13 follows trivially from our assumption that the labels assigned to a forest F_i are unique. Let $s_1, s_2, s_3, s_4 \in \mathcal{S}$. If s_1 and s_2 belong to two different groups from the same box B_i, s_1 and s_2 are not connected in F_i. If s_3 and s_4 are in the same group, s_3 and s_4 are connected in some forest, and $d(e(s_1), e(s_2))$ should therefore be different from $d(e(s_3), e(s_4))$, which cannot be the case if $e(s_1) = e(s_3)$ and $e(s_2) = e(s_4)$.

THEOREM 9. *A connectivity labeling scheme for forests with n nodes that assigns unique labels needs labels of length* $\log n + \Omega(\log \log n)$.

6.4. Sibling labeling schemes. In this section we consider the minimum label length required to answer sibling queries in a tree if the labels assigned to the nodes must be unique. We consider a forest of trees $\mathcal{F}_{\mathcal{S}}(k)$ of k trees T_i, $1 \leq i \leq k \leq \log n$. Let $B(j)$ be a complete balanced binary rooted tree with 2^j leaves and $2^{j+1}-1$ nodes.

The tree T_i consists of a tree $B = B(\log n - i)$, where each leaf from B in T_i has 2^i children. These children are the set $\mathcal{S}$. The box B_i consists of the subset of nodes from $\mathcal{S}$ which comes from T_i. The nodes in box B_i are partitioned into groups such that two nodes which belong to the same group are siblings. The first property from Lemma 13 follows trivially from our assumption that the labels assigned to a tree are unique. Let $s_1, s_2, s_3, s_4 \in \mathcal{S}$. Since s_1 and s_2 does not belong to the same group, s_1 and s_2 are not siblings. If s_3 and s_4 belongs to the same group, s_3 and s_4 are siblings. Therefore $d(e(s_1), e(s_2))$ should be different from $d(e(s_3), e(s_4))$, which cannot be the case if $e(s_1) = e(s_3)$ and $e(s_2) = e(s_4)$. The maximum degree Δ of a tree in $\mathcal{F}_{\mathcal{S}}(k)$ is 2^k, and $|\mathcal{S}| = nk$, giving the next theorem.

THEOREM 10. *A sibling labeling scheme for trees with n nodes and maximum degree Δ that assigns unique labels needs labels of length* $\log n + \Omega(\log\log\Delta)$.

6.5. 1-relationship and 2-restricted distance labeling schemes. In this section we consider the minimum label length required to answer 1-relationship and 2-restricted distance queries in a tree. To show the bound for 1-relationship labeling schemes we show that a labeling scheme for answering both parent and sibling queries needs to use labels of length $\log + \Omega(\log\log n)$. Let $\mathcal{F}_{\mathcal{SP}}$ be the forest $\mathcal{F}_{\mathcal{S}}(\log n)$ to which we have added a child to each leaf in the forest $\mathcal{F}_{\mathcal{S}}(\log n)$. We let $\mathcal{S}$ be the same subset of nodes as in the previous section. Let s_1, s_2 belong to the same box, $s_1 \neq s_2$, and let c be the child of s_1. Since s_2 is not a parent to c, s_1 and s_2 must be assigned different labels. Hence, the first property of Lemma 13 is satisfied.

THEOREM 11. *A 1-relationship labeling for trees with n nodes needs labels of length* $\log n + \Omega(\log\log n)$.

For 2-restricted distance labeling schemes we use $\mathcal{F}_{\mathcal{SP}}$ and the same partition as above. Let s_1, s_2 belong to the same box, $s_1 \neq s_2$, and let c be the child of s_1. Since the distance from s_1 to c is 1 and the distance from s_2 to c is 3, s_1 and s_2 must be assigned different labels. Furthermore, the distance between two nodes in $\mathcal{S}$ is 2 if and only if they are siblings, and by the same observations as in the sibling labeling scheme the result follows.

THEOREM 12. *A 2-restricted distance labeling scheme for trees with n nodes needs labels of length* $\log n + \Omega(\log\log n)$.

6.6. Biconnectivity labeling schemes. In this section we consider the minimum label length required to answer biconnectivity queries in a graph. Let G_i be the graph consisting of 2^i disjoint cycles $C_i = \{c_1, \ldots, c_{2^i}\}$ each of length $n/2^i$. Furthermore, for each node $v \in V(C_i)$, G_i contain two nodes $v_1, v_2 \notin V(C_i)$ connected with each other and v. Let $\mathcal{F}_{\mathcal{B}}$ be the family $G_i, 1 \leq i \leq \log n - 2$, and let $\mathcal{S}$ be the set of nodes in $C_i, 1 \leq i \leq \log n - 2$. Then $|\mathcal{S}| = n(\log n - 2)$. The box B_i is the nodes in $\mathcal{S}$ from G_i, and two nodes are in the same group if they are biconnected. Note that cycles of length less than 3 are not biconnected and therefore the restriction $i \leq \log n - 2$ is important. Let $s_1, s_2 \in \mathcal{S}$ belong to the same box, $s_1 \neq s_2$ and let v_1 and v_2 be the nodes connected to s_1 but not on the cycle containing s_2. Since v_1 and v_2 are biconnected with s_1 but not s_2, $e(s_1) \neq e(s_2)$. Let $s_1, s_2, s_3, s_4 \in \mathcal{S}$, where s_1 and s_2 belong to different groups in the same box. This implies that s_1 and s_2 are not biconnected and if $e(s_1) = e(s_3)$ and $e(s_2) = e(s_4)$, s_3 and s_4 must also belong to different groups.

THEOREM 13. *A biconnectivity labeling scheme for graphs with n nodes needs labels of length* $\log n + \Omega(\log\log n)$.

Acknowledgments. We thank the anonymous reviewers for the very insightful and useful comments and Inge Li Gørtz for proofreading.

REFERENCES

[1] S. Abiteboul, H. Kaplan, and T. Milo, *Compact labeling schemes for ancestor queries*, in Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms, 2001, pp. 547–556.

[2] S. Alstrup, C. Gavoille, H. Kaplan, and T. Rauhe, *Nearest common ancestors: A survey and a new distributed algorithm*, in Proceedings of the 14th Annual ACM Symposium on Parallel Algorithms and Architecture, 2002.

[3] S. Alstrup and T. Rauhe, *Improved labeling schemes for ancestor queries*, in Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms, 2002.

[4] S. Alstrup and T. Rauhe, *Small induced universal graphs and compact implicit graph representations*, in Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002.

[5] M. A. Breuer, *Coding vertexes of a graph*, IEEE Trans. Inform. Theory, 12 (1966), pp. 148–153.

[6] M. A. Breuer and J. Folkman, *An unexpected result on coding vertices of a graph*, J. Math. Anal. Appl., 20 (1967), pp. 583–600.

[7] E. Cohen, E. Halperin, H. Kaplan, and U. Zwick, *Reachability and distance queries via 2-hop labels*, in Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms, 2002.

[8] E. Cohen, H. Kaplan, and T. Milo, *Labeling dynamic XML trees*, in Proceedings of the 21st Annual ACM Symposium on Principles of Database Systems, 2002.

[9] R. Diestel, *Graph Theory*, Springer-Verlag, New York, 2000.

[10] C. Gavoille, M. Katz, N. Katz, C. Paul, and D. Peleg, *Approximate distance labeling schemes*, in Proceedings of the 9th Annual European Symposium on Algorithms, Lecture Notes in Comput. Sci. 2161, Springer-Verlag, New York, 2001, pp. 476–488.

[11] C. Gavoille and C. Paul, *Split decomposition and distance labeling: An optimal scheme for distance hereditary graphs*, in Proceedings of the 9th European Conference on Combinatorics, Graph Theory and Applications, 2001.

[12] C. Gavoille and D. Peleg, *Compact and localized distributed data structures*, Distributed Computing, 16 (2003), pp. 111–120.

[13] C. Gavoille, D. Peleg, S. Perennes, and R. Raz, *Distance labeling in graphs*, in Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms, 2001.

[14] D. Harel and R. E. Tarjan, *Fast algorithms for finding nearest common ancestors*, SIAM J. Comput., 13 (1984), pp. 338–355.

[15] D. A. Huffman, *A method for construction of minimum-redundancy codes*, in Proceedings of the Institute of Radio Engineers, 1952.

[16] S. Kannan, M. Naor, and S. Rudich, *Implicit representation of graphs*, in Proceedings of 20th Annual ACM Symposium on Theory of Computing, 1988.

[17] S. Kannan, M. Naor, and S. Rudich, *Implicit representation of graphs*, SIAM J. Discrete Math., 5 (1992), pp. 596–603.

[18] H. Kaplan and T. Milo, *Short and simple labels for small distances and other functions*, in Proceedings of the 7th Workshop on Algorithms and Data Structures, Lecture Notes in Comput. Sci. 2125, Springer-Verlag, New York, 2001.

[19] H. Kaplan, T. Milo, and R. Shabo, *A comparison of labeling schemes for ancestor queries*, in Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms, 2002.

[20] M. Katz, N. Katz, and D. Peleg, *Distance labeling schemes for well-separated graph classes*, in Proceedings of the 17th Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Comput. Sci. 1170, Springer-Verlag, New York, 2000.

[21] M. Katz, N. A. Katz, A. Korman, and D. Peleg, *Labeling schemes for flow and connectivity*, in Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms, 2002.

[22] D. Peleg, *Proximity-preserving labeling schemes and their applications*, in Graph-Theoretic Concepts in Computer Science, 25th International Workshop, Lecture Notes in Comput. Sci. 1665, Springer-Verlag, New York, 1999, pp. 30–41.

[23] N. Santoro and R. Khatib, *Labeling and implicit routing in networks*, Comput. J., 28 (1985), pp. 5–8.

[24] R. E. Tarjan, *Depth-first search and linear graph algorithms*, SIAM J. Comput., 1 (1972), pp. 146–160.

[25] M. Thorup and U. Zwick, *Approximate distance oracles*, in Proceedings of the 13th Annual ACM Symposium on Theory of Computing, 2001, pp. 1–10.

[26] M. Thorup and U. Zwick, *Compact routing schemes*, in Proceedings of the 13th Annual ACM Symposium on Parallel Algorithms and Architecture, Vol. 13, 2001.

[27] A. K. Tsakalidis, *Maintaining order in a generalized linked list*, Acta Inform., 21 (1984), pp. 101–112.

SIAM J. DISCRETE MATH.
Vol. 19, No. 2, pp. 463–480

A CLASS OF GENERAL SUPERTREE METHODS FOR NESTED TAXA*

PHILIP DANIEL[†] AND CHARLES SEMPLE[†]

Abstract. Amalgamating smaller evolutionary trees into a single parent tree is an important task in evolutionary biology. Traditionally, the (supertree) methods used for this amalgamation take a collection of leaf-labeled trees as their input. However, it has been recently highlighted that, in practice, such an input is somewhat limiting and that one would like supertree methods for collections of trees in which some of the interior vertices, as well as all of the leaves, are labeled [R. D. M. Page, in *Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life*, O. Bininda-Emonds, ed., Kluwer, Dordrecht, The Netherlands, 2004, pp. 247–265]. In this paper, we describe what appears to be the first approach for constructing such methods and show that any method using this approach satisfies particular desirable properties.

Key words. rooted phylogenetic tree, rooted semilabeled tree, nested taxa, supertree method, supertree

AMS subject classifications. 05C05, 92D15

DOI. 10.1137/S0895480104441462

1. Introduction.

In evolutionary biology, *supertree methods* have become a fundamental process for constructing an evolutionary tree that best represents the information exhibited by the original input. These methods amalgamate an input collection of smaller evolutionary trees on overlapping sets of taxa into a single parent tree called a *supertree*. The increasing popularity of supertree methods is highlighted by a recent survey [3] and a published book [4].

If the input collection of trees carries no conflicting information, then one would like the resulting supertree to preserve all of the ancestral relationships displayed by each of the trees in this collection. For collections of rooted phylogenetic trees, there is a polynomial-time algorithm that finds such a tree. In practice, however, incompatibility is more common and so one seeks a method that resolves these conflicts in a sensible way, while still producing a supertree that has a number of attractive properties. The following list of desirable properties for any supertree method applied to a collection $\mathcal{P}$ of rooted phylogenetic trees is given in [12]:

(i) The method runs in polynomial time in the size of the input.

(ii) The resulting supertree displays all rooted binary subtrees shared by all of the trees in $\mathcal{P}$.

(iii) If $\mathcal{P}$ is compatible, then the resulting supertree displays each of the trees in $\mathcal{P}$.

(iv) The method satisfies the following two natural symmetry properties of *ordering* and *renaming*:

  (a) The resulting supertree is independent of the order in which the members of $\mathcal{P}$ are listed.

*Received by the editors February 26, 2004; accepted for publication (in revised form) February 2, 2005; published electronically November 4, 2005. The first author was supported by the New Zealand Institute of Mathematics and Its Applications funded program Phylogenetic Genomics, and the second author was supported by the New Zealand Marsden Fund (UOC310).

http://www.siam.org/journals/sidma/19-2/44146.html

[†]Biomathematics Research Centre, Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand (pjd62@student.canterbury.ac.nz, c.semple@math.canterbury.ac.nz).

(b) If we rename all the species and then apply the method to this new collection of input trees, the resulting supertree tree is the one obtained by applying the method to the original collection $\mathcal{P}$, but with the species renamed as before.

(v) The method allows a possible weighting of the trees in $\mathcal{P}$.

To date, the algorithms MinCutSupertree [12] and its modified version [9] are the only two supertree methods for rooted phylogenetic trees that have been shown to satisfy all the above properties. We remark here that (iv) may seem trivial to satisfy, but for collections of unrooted phylogenetic trees, it has been shown that no supertree method for such collections can simultaneously satisfy (iii) and both parts of (iv) [14].

In this paper, we present a general supertree method for collections of rooted semilabeled trees, that is, rooted trees in which some (possibly none) of the interior vertices as well as all of the leaves are labeled. Making the extension from rooted phylogenetic trees to rooted semilabeled trees means that we allow nested taxa in the input. In particular, the interior labels represent taxa at a higher taxonomic level than any of their descendants, for example, families versus genera or genera versus species. One of the main features of this supertree method is that it purposely allows for the possibility of variants. Indeed, provided the input satisfies two natural ancestor-descendant pairwise properties, any such variant constructed from it satisfies all the rooted semilabeled tree analogues of the desirable properties (i)–(iii) above. Moreover, although the rooted semilabeled tree analogues of (iv) and (v) are dependent on the constructed variant, satisfying these additional properties is not difficult. We highlight this with an example of such an algorithm. To the best of our knowledge, this is the first time such supertree methods for rooted semilabeled trees have been considered. The next section contains some further background and necessary preliminaries for the rest of the paper.

2. Background and preliminaries. Throughout this paper, we will assume that the reader has some familiarity with the basics of phylogenetics. Unless otherwise stated, the notation and terminology follows Semple and Steel [13].

A *rooted phylogenetic tree (on X)* is an ordered pair $(T;\phi)$ consisting of a rooted tree T in which all interior vertices have degree at least three except the root, which has degree at least two and a bijective map ϕ from X to the leaf set of T. Rooted phylogenetic trees on X are also called *rooted phylogenetic X-trees.* Loosely speaking, a rooted phylogenetic X-tree is a rooted tree whose leaves are bijectively labeled with the elements of X. The leftmost tree in Figure 1 is an example of a rooted phylogenetic tree, where $X = \{a, b, c, d\}$.

Let $\mathcal{T}'$ be a rooted phylogenetic tree on X', and let X be a subset of X'. The *restriction* of $\mathcal{T}'$ to X is the rooted phylogenetic tree that is obtained from the minimal rooted subtree of $\mathcal{T}'$ induced by the elements of X by suppressing all nonroot vertices of degree two. This restriction is denoted by $\mathcal{T}'|X$. We say that $\mathcal{T}'$ *displays* a rooted phylogenetic X-tree $\mathcal{T}$ if, up to isomorphism, $\mathcal{T}'|X$ is a refinement of $\mathcal{T}$. Intuitively, $\mathcal{T}'$ displays $\mathcal{T}$ if $\mathcal{T}'$ preserves all of the ancestral relationships described by $\mathcal{T}$. The reason for allowing refinement is that, from a biological viewpoint, vertices of outdegree at least three usually represent an uncertainty to the exact order of speciation as oppose to a multiple speciation event. A collection $\mathcal{P}$ of rooted phylogenetic trees is said to be *compatible* if there exists a rooted phylogenetic tree that displays each of the trees in $\mathcal{P}$. Again, intuitively, $\mathcal{P}$ is compatible if it carries no conflicting information.

Traditionally, supertree methods have been applied to rooted phylogenetic trees. One of the first such methods is Build [2]. This polynomial-time algorithm takes a

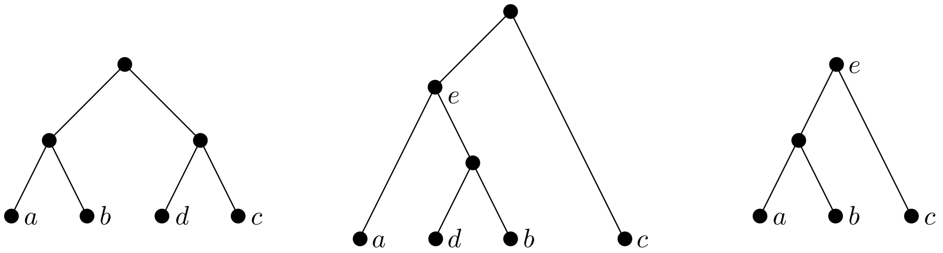


FIG. 1. *A collection $\mathcal{P}$ of rooted semilabeled trees.*

collection of rooted phylogenetic trees and determines if they are compatible, in which case it outputs a tree that displays each of the trees in the collection. Algorithms like BUILD are all-or-nothing algorithms as they return a tree only if the input data meet some criteria. However, despite this limitation, such algorithms give valuable insight into more general supertree methods. Indeed, the algorithm MINCUTSUPERTREE and its modified version is based on BUILD.

For nested taxa, the analogues of rooted phylogenetic trees and compatibility are rooted semilabeled trees and ancestral compatibility. A *rooted semilabeled tree (on X)* is an ordered pair $(T;\phi)$ consisting of a rooted tree T with vertex set V and root vertex ρ and a map $\phi : X \to V$ with the properties that for all $v \in V - \{\rho\}$ of degree at most two, $v \in \phi(X)$ and, if ρ has degree zero or one, then $\rho \in \phi(X)$. Rooted semilabeled trees on X are also called *rooted X-trees*. Furthermore, if ϕ is one-to-one, then $(T;\phi)$ is said to be *singularly labeled*. Observe that the definition of rooted semilabeled trees extends the definition of rooted phylogenetic trees by allowing (i) some of the interior (nonleaf) vertices as well as all the leaves to be labeled by the elements of X and (ii) vertices may be labeled by more than one element of X. Examples of rooted semilabeled trees that are singularly labeled are shown in Figure 1.

Let $X \subseteq X'$ and let $a, b \in X$. A rooted X'-tree $\mathcal{T}'$ *ancestrally displays* a rooted X-tree $\mathcal{T}$ if $\mathcal{T}'|X$ refines $\mathcal{T}$ so that whenever a is a strict descendant of b in $\mathcal{T}$, a is a strict descendant of b in $\mathcal{T}'|X$. The formal definition of strict descendant is given at the end of this section, but intuition should suffice for the moment. A collection $\mathcal{P}$ of rooted semilabeled trees is *ancestrally compatible* if there is a rooted semilabeled tree $\mathcal{T}$ that ancestrally displays each of the trees in $\mathcal{P}$, in which case we say that $\mathcal{T}$ *ancestrally displays* $\mathcal{P}$. Observe that if $\mathcal{P}$ consists of rooted phylogenetic trees and is compatible, then $\mathcal{P}$ is ancestrally compatible as none of the trees in $\mathcal{P}$ contains any interior labels. Conversely, suppose that $\mathcal{P}$ is ancestrally compatible and consists of rooted phylogenetic trees. Let $\mathcal{T}$ be a rooted semilabeled tree that ancestrally displays $\mathcal{P}$. Let $\mathcal{T}'$ be the rooted phylogenetic tree that is obtained from $\mathcal{T}$ by replacing each interior label x with a pendant edge joining the interior vertex previously labeled by x and labeling the other end-vertex x. It is now easily checked that $\mathcal{T}'$ displays $\mathcal{P}$.

Page [10] recently motivated the problem of developing supertree methods for nested taxa and initially posed the problem of constructing a polynomial-time algorithm for determining the ancestral compatibility of an arbitrary collection of rooted semilabeled trees. In answer to this problem, Daniel and Semple [7] presented an algorithm called ANCESTRALBUILD. Analogous to BUILD, this polynomial-time algorithm is an all-or-nothing algorithm and determines if a collection $\mathcal{P}$ of rooted semilabeled trees are ancestrally compatible, in which case it outputs a rooted semilabeled tree that ancestrally displays $\mathcal{P}$. With ANCESTRALBUILD in hand, the next natural step forward is to construct a more general supertree method for rooted semilabeled trees.

In section 3 of this paper, we present a supertree method for collections of rooted semilabeled trees that are singularly labeled. Called NestedSupertree, this method outputs either a rooted semilabeled tree or a statement indicating that either there is a pair of taxa that are not pairwise consistent or there is an ancestor-descendant contradiction. Strictly speaking, this is still an all-or-nothing algorithm. However, such an inconsistency or a contradiction is very particular, and one that we believe in practice could be resolved separately. Based on AncestralBuild, one of the attractions of NestedSupertree is that it is able to be easily refined to give rise to a number of possible variants, each of which is a supertree method for rooted semilabeled trees that are singularly labeled. Moreover, we show in sections 3 and 4 that any such variant satisfies all the rooted semilabeled tree analogues of properties (i)–(iii) in the introduction. Furthermore, in section 5, we describe one particular variant where the rooted semilabeled trees in the input are weighted. In addition to (i)–(iii), the resulting algorithm satisfies the rooted semilabeled tree analogues of (iv) and (v). The restriction to collections of rooted semilabeled trees that are singularly labeled is for simplicity and functionality (see remarks in section 3). Indeed, in practice, this is not much of a restriction as rooted semilabeled trees are typically singularly labeled.

In the final section of this paper, section 6, we consider what happens when NestedSupertree is applied to a collection $\mathcal{P}$ of rooted phylogenetic trees. In this case, the minor conditions on $\mathcal{P}$ referred to above are redundant and that NestedSupertree applied to $\mathcal{P}$ always returns a rooted phylogenetic tree. We show that if $\mathcal{P}$ is compatible, then the rooted phylogenetic tree returned by NestedSupertree is the same as that returned by Build. Thus NestedSupertree is a generalization of Build. In fact, as we will see, it also generalizes AncestralBuild in a corresponding way.

Before ending this section with some preliminaries we make two comments. First, in addition to the properties listed in the introduction, one other property is given in [12]. This property says that "the resulting supertree displays all 'nestings' shared by all of the trees in $\mathcal{P}$," where one subset of the labels in $\mathcal{P}$ *nests in* another if the most recent common ancestor of the former is a strict descendant of the most recent common ancestor of the latter. It has been recently shown by Willson [15] that the proof in [12] that establishes MinCutSupertree has this property is incorrect and, in fact, that MinCutSupertree does not have this property. (We note that if one adds the condition that "A is a subset of B" in the statement associated with this proof, then the proof is correct and MinCutSupertree is guaranteed to have the nesting property provided the first set of labels is a subset of the other.) Whether displaying all shared nestings of the input collection is a desirable property is debatable. We simply note here that NestedSupertree also does not have this property. For the curious reader, there is a general supertree method for collections $\mathcal{P}$ of rooted phylogenetic trees that satisfies this nesting property as well as the properties listed in the introduction. In particular, first use the Build algorithm to either produce a supertree that displays $\mathcal{P}$, in which case the supertree method outputs this tree, or recognize that $\mathcal{P}$ is not compatible. If the latter happens, construct the "Adams consensus tree" $\mathcal{T}$ (see [1] or [13]) for the set $\mathcal{P}'$ of rooted phylogenetic trees obtained from $\mathcal{P}$ by restricting each tree to the subset of labels of $\mathcal{P}$ that are common to each tree in $\mathcal{P}$. This tree displays all of the nestings shared by all of the trees in $\mathcal{P}'$ and hence $\mathcal{P}$. Now, for each remaining label a in $\mathcal{P}$, adjoin a to the root of $\mathcal{T}$ with a distinct new edge. The supertree method outputs the resulting tree. Second, the approach taken by NestedSupertree and the approach of MinCutSupertree are very different. A comparison between these two methods for rooted phylogenetic trees would make an interesting project.

Finally, some preliminaries. Typically, one views a rooted tree as an undirected graph. However, it will often be convenient in this paper to view a rooted tree as a directed graph where each edge is replaced with an arc directed away from the root. Now let $\mathcal{T} = (T;\phi)$ be a rooted semilabeled tree on X. The set X is called the *label set* of $\mathcal{T}$ and the elements of X are called *labels*. We also use $\mathcal{L}(\mathcal{T})$ to denote the label set of $\mathcal{T}$. If v is a vertex of T, we say that the elements of $\phi^{-1}(v)$ *label* v. Furthermore, $\mathcal{T}$ is *fully labeled* if every vertex of T is labeled by an element of X. For a collection $\mathcal{P}$ of rooted semilabeled trees, we denote the union of the label sets of the trees in $\mathcal{P}$ by $\mathcal{L}(\mathcal{P})$. Moreover, we call an element x of $\mathcal{L}(\mathcal{P})$ *common* if $x \in \bigcap_{\mathcal{T}\in\mathcal{P}} \mathcal{L}(\mathcal{T})$.

There is a natural and useful partial order on the label set $\mathcal{L}(\mathcal{T})$ of a rooted semilabeled tree $\mathcal{T} = (T;\phi)$. This partial order is obtained by setting $b \leq_{\mathcal{T}} a$ if the path from the root of T to $\phi(a)$ includes $\phi(b)$, in which case we say that a is a *descendant* of b. If $b <_{\mathcal{T}} a$, then we say that a is a *strict descendant* of b. Furthermore, $a, b \in \mathcal{L}(\mathcal{T})$ are *not comparable* under $\leq_{\mathcal{T}}$ if neither $b \leq_{\mathcal{T}} a$ nor $a \leq_{\mathcal{T}} b$ holds. Essentially, a and b are not comparable in $\mathcal{T}$ if a is not a descendant of b and b is not a descendant of a. In Figure 1, e and c are not comparable in the middle tree, but c is a (strict) descendant of e in the rightmost tree.

Last, a *rooted triple* is a rooted phylogenetic tree that has two interior vertices and whose label set has size three. We denote the rooted triple $\mathcal{T}$ with label set $\{a,b,c\}$ by $ab|c$ if the path from a to b does not intersect the path from the root to c. For a collection $\mathcal{P}$ of rooted semilabeled trees, a rooted triple whose label set $\{a,b,c\}$ is a subset of $\bigcap_{\mathcal{T}\in\mathcal{P}} \mathcal{L}(\mathcal{T})$ is *common* relative to $\mathcal{P}$ if, for all $\mathcal{T}_1, \mathcal{T}_2 \in \mathcal{P}$, $\mathcal{T}_1|\{a,b,c\}$ is isomorphic to $\mathcal{T}_2|\{a,b,c\}$. Note that none of a, b, c need label a leaf of $\mathcal{T}_1$ or $\mathcal{T}_2$. The rooted triple $ab|c$ is common to the three rooted semilabeled trees shown in Figure 1.

3. The algorithm NestedSupertree**.** For a collection $\mathcal{P}$ of rooted semilabeled trees that are singularly labeled, the algorithm NestedSupertree applied to $\mathcal{P}$ is based on a particular construction and two graphs. We describe the construction first and then the two graphs.

Let $\mathcal{T} = (T;\phi)$ be a rooted semilabeled tree on X, where T has vertex set V. We say that a rooted fully labeled tree $\mathcal{T}_1 = (T;\phi_1)$ on X_1, where $X \subseteq X_1$, has been obtained from $\mathcal{T}$ by *adding distinct new labels* if for all distinct $u, v \in V$, the following properties are satisfied:

1. If $\phi^{-1}(v)$ is nonempty, then $\phi_1^{-1}(v) = \phi^{-1}(v)$.
2. If $\phi^{-1}(v)$ is empty, then $|\phi_1^{-1}(v)| = 1$.
3. If $\phi^{-1}(u)$ and $\phi^{-1}(v)$ are both empty, then $\phi_1^{-1}(u) \neq \phi_1^{-1}(v)$.

Intuitively, $\mathcal{T}_1$ has been obtained from $\mathcal{T}$ by adding a distinct new label to each nonlabeled vertex of $\mathcal{T}$. For a collection $\mathcal{P}$ of rooted semilabeled trees, we say that $\mathcal{P}_1$ has been obtained from $\mathcal{P}$ by adding distinct new labels if it has been obtained by adding distinct new labels to each tree in $\mathcal{P}$ so that no pair of added labels are the same. Although NestedSupertree is applied to $\mathcal{P}$, all the work in the method goes into constructing a supertree for a collection of rooted fully labeled trees that has been obtained from $\mathcal{P}$ by adding distinct new labels.

We now describe the two graphs each of which consists of both arcs (directed edges) and edges. For the purposes of this paper and to avoid confusion, we will call a graph that contains both arcs and edges a *mixed* graph. Let $\mathcal{P}$ be a collection of rooted semilabeled trees and let $\mathcal{P}'$ be a collection of rooted fully labeled trees obtained from

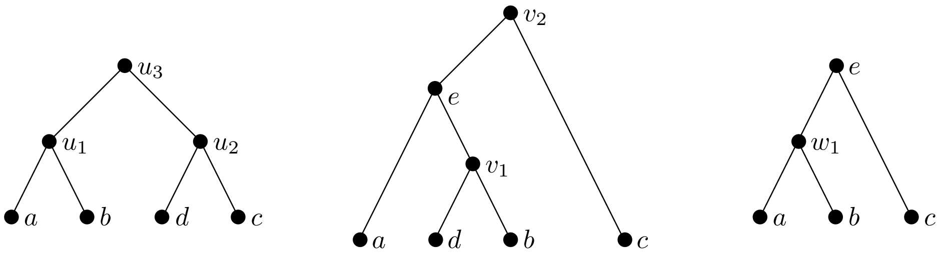


FIG. 2. *A collection $\mathcal{P}'$ of rooted fully labeled trees.*

$\mathcal{P}$ by adding distinct new labels. The *descendancy graph of* $\mathcal{P}'$, denoted $D(\mathcal{P}')$, is the mixed graph whose vertex set is $\mathcal{L}(\mathcal{P}')$, whose arc set is

$$\{(c, a) : c <_{\mathcal{T}} a \text{ for some } \mathcal{T} \in \mathcal{P}'\},$$

and whose edge set is

$$\{\{a, b\} : a \text{ is not comparable to } b \text{ under } \leq_{\mathcal{T}} \text{ for some } \mathcal{T} \in \mathcal{P}'\}.$$

The descendancy graph is said to be *acyclic* if, ignoring edges, it has no directed cycles.

The second graph $D'(\mathcal{P}')$ is obtained from the descendancy graph $D(\mathcal{P}')$ of $\mathcal{P}'$ as follows. For each common rooted triple $a_1a_2|b$ of $\mathcal{P}$, add a new vertex labeled $\overline{a_1a_2|b}$, a new arc from $\overline{a_1a_2|b}$ to a_1, and a new arc from $\overline{a_1a_2|b}$ to a_2. Vertices of the form $\overline{a_1a_2|b}$ are called *rooted triple vertices* of $D'(\mathcal{P}')$; all other vertices of $D'(\mathcal{P}')$ are called *label vertices.* We call $D'(\mathcal{P}')$ the *modified descendancy graph* of $\mathcal{P}'$.

In general, let G be a mixed graph and let G' be the directed graph obtained from G by deleting all of the edges in the edge set of G. Thus the arc set of G' is equal to the arc set of G. A vertex v of G has *indegree zero* if v has indegree zero in G'. Similarly, a subset of the vertex set of G is the vertex set of an *arc component* of G if it is the vertex set of a component of G'. Furthermore, for a subset V_1 of the vertex set of G, the restriction of G to V_1 is the subgraph of G that is obtained by deleting all vertices not in V_1 together with their incident edges and arcs. This restriction is denoted by $G|V_1$.

Example 3.1. To illustrate the above construction and mixed graphs, let $\mathcal{P}$ be the collection of rooted semilabeled trees shown in Figure 1 and let $\mathcal{P}'$ be the collection of rooted fully labeled trees obtained from $\mathcal{P}$ by adding distinct new labels as shown in Figure 2.

The modified descendancy graph of $\mathcal{P}'$ is shown in Figure 3, where, for simplicity, the edges as well as the arcs (c, a) where a is not an immediate descendant of c are omitted. If these edges were included, there would, for example, be an edge joining the label vertices w_1 and c as they are not comparable in the rightmost tree of Figure 2. Furthermore, to highlight the one rooted triple vertex, its outgoing arcs are drawn as dashed arrows. This example will be referred to later in this section and also in section 5.

Last, let $\mathcal{P}$ be a collection of rooted semilabeled trees that are singularly labeled, and let a and b be elements of $\mathcal{L}(\mathcal{P})$. We say that a and b are *pairwise consistent* if, whenever a is a strict descendant of b in some tree in $\mathcal{P}$, a is always a strict descendant of b in every tree of $\mathcal{P}$ whose label set contains both a and b. Furthermore, $\mathcal{P}$ is said to be *pairwise consistent* if all pairs of labels in $\mathcal{L}(\mathcal{P})$ are pairwise consistent.

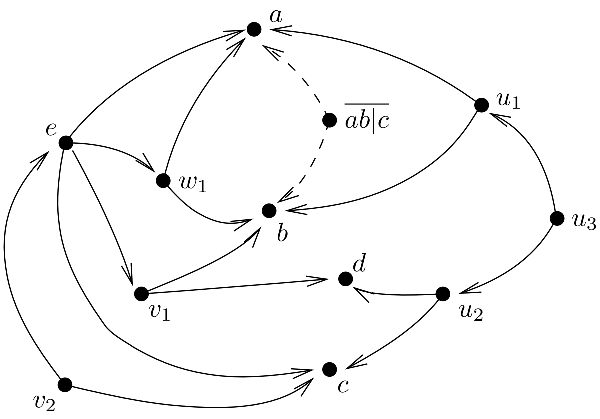


FIG. 3. *The modified descendancy graph of* $\mathcal{P}'$.

We now describe NESTEDSUPERTREE and its subroutine DESCENDANT. An illustrative example and some informative remarks follow these descriptions. In brief, NESTEDSUPERTREE constructs a rooted semilabeled tree by starting at the root and working downwards toward the leaves. The main workings of the method are contained within a subroutine called DESCENDANT. This subroutine uses successive restrictions of a certain modified descendancy graph to determine how this rooted semilabeled tree is constructed.

ALGORITHM NESTEDSUPERTREE($\mathcal{P}$).

Input: A collection $\mathcal{P}$ of rooted semilabeled trees that are singularly labeled.

Output: A rooted semilabeled tree $\mathcal{T}$ with label set $\mathcal{L}(\mathcal{P})$, the statement $\mathcal{P}$ *is not pairwise consistent*, or the statement $\mathcal{P}$ *has an ancestor-descendant contradiction*.

1. For each pair $a, b \in \mathcal{P}$, check that a and b are pairwise consistent. If not, then halt and return $\mathcal{P}$ *is not pairwise consistent.*
2. Construct a collection $\mathcal{P}'$ of rooted fully labeled trees from $\mathcal{P}$ by adding distinct new labels.
3. Construct the descendancy graph $D(\mathcal{P}')$ of $\mathcal{P}'$.
4. If $D(\mathcal{P}')$ has a directed cycle, then halt and return $\mathcal{P}$ *has an ancestor-descendant contradiction.*
5. Construct the modified descendancy graph $D'(\mathcal{P}')$ of $\mathcal{P}'$.
6. Call the subroutine DESCENDANT($D'(\mathcal{P}'), v'$).
7. Remove the added labels from $\mathcal{T}'$ (the rooted semilabeled tree outputted by DESCENDANT), suppress any resulting unlabeled vertex that has indegree one and outdegree one, and, if the root is unlabeled and has degree one, relocate the root to the nearest vertex that is either labeled or has outdegree at least two. Return the resulting rooted semilabeled tree $\mathcal{T}$.

ALGORITHM DESCENDANT($D'(\mathcal{P}'), v'$).

Input: A graph $D'(\mathcal{P}')$.

Output: A rooted fully labeled tree $\mathcal{T}'$ with root vertex v'.

1. Let $\mathcal{S}_0$ denote the set of label vertices of $D'(\mathcal{P}')$ that have indegree zero and no incident edges.
2. If $\mathcal{S}_0$ is empty, then choose $\mathcal{S}_0$ to be any nonempty subset of label vertices of $D'(\mathcal{P}')$ that have indegree zero.
3. Delete the elements of $\mathcal{S}_0$ (and their incident arcs and their incident edges) from

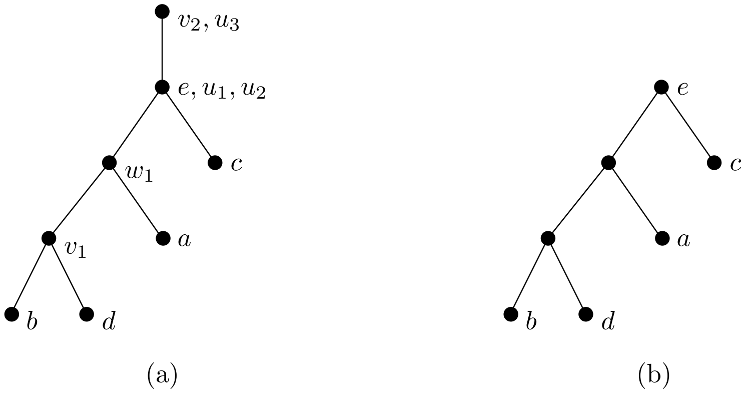


FIG. 4. (a) *One possible output of* DESCENDANT *when applied to* $D'(\mathcal{P}')$ *and* (b) *the corresponding output of* NESTEDSUPERTREE.

$D'(\mathcal{P}')$. Furthermore, for each common rooted triple $a_1a_2|b$ of $\mathcal{P}$, delete the rooted triple vertex $\overline{a_1a_2|b}$ if, in the resulting mixed graph, the arc component containing a_1 and a_2 does not contain the label vertex b.

4. Let $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k$ denote the vertex sets of the arc components of the graph obtained at the end of step 3.
5. For each element $i \in \{1, 2, \ldots, k\}$, call DESCENDANT($D'(\mathcal{P}')|\mathcal{S}_i, v'_i$). Assign the labels in $\mathcal{S}_0$ to v' and attach $\mathcal{T}'_i$ to v' via the edge $\{v'_i, v'\}$.

Example 3.2. As an example of NESTEDSUPERTREE applied to a collection of rooted semilabeled trees that are singularly labeled, let $\mathcal{P}$ and $\mathcal{P}'$ be the collections described in Example 3.1. On the first iteration of DESCENDANT, the label vertices v_2 and u_3 in the modified descendancy graph $D'(\mathcal{P}')$ have indegree zero and no incident edges, and no other label vertices have this property. Therefore, in this iteration, $\mathcal{S}_0 = \{v_2, u_3\}$. Furthermore, the graph obtained from $D'(\mathcal{P}')$ by deleting the elements of $\mathcal{S}_0$ has exactly one arc component.

In the second iteration, the label vertices of the inputted graph that have indegree zero are e, u_1, and u_2, and each of these has an incident edge. Therefore, in this iteration, we can choose any nonempty subset of $\{e, u_1, u_2\}$ to be $\mathcal{S}_0$. If we choose $\mathcal{S}_0$ to be the whole set, then, in all subsequent iterations of the algorithm, there is always a nonempty set of label vertices of the corresponding graph that have indegree zero and no incident edges. By making this choice, DESCENDANT eventually returns the rooted fully labeled tree shown in Figure 4(a) and NESTEDSUPERTREE returns the rooted semilabeled tree shown in Figure 4(b).

Remarks.

1. Observe that in step 2 of DESCENDANT a choice can be made on the make up of $\mathcal{S}_0$. This is the part of the algorithm that allows for variants. One possible way to make this choice is described in section 5. Note that, as we will soon see, if the subroutine is called but step 2 is never invoked, then the supertree returned by NESTEDSUPERTREE ancestrally displays $\mathcal{P}$.

2. One of the attractions of a general supertree algorithm is that conflicts are resolved in some way so that one always outputs a supertree whether or not the original collection of input trees is compatible. In the case that the input is a collection of rooted phylogenetic trees, it is reasonable that any supertree algorithm resolves such conflicts. However, in the case that the input is a collection of rooted semilabeled

trees, it appears to us that there are some fundamental ancestor-descendant conflicts that should be resolved separately. Two such conflicts are when $\mathcal{P}$ is not pairwise consistent or has an ancestor-descendant contradiction. Finding such conflicts can be easily done in polynomial time. In the case of ancestor-descendant contradictions, see the proof of Lemma 3.4.

3. Rooted triple vertices and associated arcs are added to the descendancy graph of $\mathcal{P}'$ so that any tree outputted by NestedSupertree preserves all the common rooted triples of $\mathcal{P}$. This property and, in particular, the rooted semilabeled tree analogue of desirable property (ii) are established in the next section.

4. Proposition 3.6 shows that, provided $\mathcal{P}$ is pairwise consistent and the descendancy graph of $\mathcal{P}'$ is acyclic, NestedSupertree returns a rooted semilabeled tree. Thus we can always find a nonempty set $\mathcal{S}_0$ as described in steps 1 and 2 of the subroutine Descendant.

5. Last, the check for the pairwise consistency of $\mathcal{P}$ and the restriction that each tree in the input collection is singularly labeled could be removed from NestedSupertree. However, if either is done, then there is no guarantee that the resulting supertree satisfies the rooted semilabeled tree analogue of (ii) in the introduction.

The rest of this section establishes some basic properties of NestedSupertree, in particular, the rooted semilabeled tree analogues of (i) (Proposition 3.7) and (iii) (Proposition 3.3). Further properties are established in the next section.

We begin by making the following observation. Recall from the introduction that AncestralBuild is a polynomial-time algorithm that determines if a collection of rooted semilabeled trees is ancestrally compatible, in which case such a tree is returned [7]. The description of NestedSupertree closely resembles the description of AncestralBuild. Indeed, the latter can be essentially obtained from the former as follows. Remove steps 1, 4, and 5 in NestedSupertree; replace the modified descendancy graph of $\mathcal{P}'$ with the descendancy graph of $\mathcal{P}'$ in Descendant; remove the second sentence of step 3 of Descendant; and replace step 2 of Descendant "If $\mathcal{S}_0$ is empty, halt and return $\mathcal{P}'$ *is not ancestrally compatible*", in which case $\mathcal{P}$ is not ancestrally compatible. It follows that NestedSupertree can be viewed as a generalization of AncestralBuild. Indeed, we have the following proposition.

Proposition 3.3. *Let $\mathcal{P}$ be a collection of rooted semilabeled trees that are singularly labeled, and suppose $\mathcal{P}$ is ancestrally compatible. Then* NestedSupertree *applied to $\mathcal{P}$ returns a rooted semilabeled tree that ancestrally displays $\mathcal{P}$.*

Proof. Let $\mathcal{P}'$ be a collection of rooted fully labeled trees that is obtained from $\mathcal{P}$ by adding distinct new labels. Since $\mathcal{P}$ is ancestrally compatible, $\mathcal{P}$ is pairwise consistent and $D(\mathcal{P}')$ has no directed cycles. It now follows from the description of how AncestralBuild can be obtained from NestedSupertree that NestedSupertree applied to $\mathcal{P}$ returns a rooted semilabeled tree that ancestrally displays $\mathcal{P}$. □

The next two lemmas are needed for the proofs of Propositions 3.6 and 3.7. The first lemma is well known and is an easy exercise. However, we include its proof as it indicates how one can find all the ancestor-descendant contradictions of a collection of rooted semilabeled trees.

Lemma 3.4. *Let D be a connected digraph that contains no directed cycle. Then there exists a vertex of D whose indegree is zero.*

Proof. Assume no vertex of D has indegree zero. Let D' be the digraph obtained from D by reversing the orientation of the arcs of D. By assumption, every vertex of D' has outdegree at least one. Let u be a vertex of D'. Starting at u, construct a directed walk. Since each vertex of D' has an outgoing arc, we must eventually meet a vertex on this walk that has already been traversed. In particular, this means that D' contains a directed cycle, which in turn implies that D contains a directed cycle. This contradiction completes the proof of the lemma. □

LEMMA 3.5. *Let $\mathcal{P}$ be a collection of rooted fully labeled trees that are singularly labeled. Let $b \in \bigcap_{\mathcal{T}\in\mathcal{P}} \mathcal{L}(\mathcal{T})$, and let $x, y \in \mathcal{L}(\mathcal{P})$. Suppose that b is pairwise consistent with each of the labels in $\mathcal{L}(\mathcal{P})$. Then the following hold:*

(i) *If there is a directed path from b to x in $D(\mathcal{P})$, then there is an arc from b to x in $D(\mathcal{P})$. Furthermore, if there is a directed path from x to b in $D(\mathcal{P})$, then there is an arc from x to b in $D(\mathcal{P})$.*

(ii) *Suppose that (b, x) is an arc in $D(\mathcal{P})$. If (y, x) is also an arc in $D(\mathcal{P})$ and $b \neq y$, then either there is an arc from y to b in $D(\mathcal{P})$ or there is an arc from b to y in $D(\mathcal{P})$.*

Proof. We first prove (i). Assume that $by_1y_2\cdots y_kx$ is a directed path in $D(\mathcal{P})$ from b to x. As y_1 is a strict descendant of b in some tree in $\mathcal{P}$ and b is pairwise consistent with y_1, it follows that whenever b and y_1 are labels of some tree in $\mathcal{P}$, y_1 is a strict descendant of b. Since there is an arc from y_1 to y_2, there is a tree $\mathcal{T}_1$ in $\mathcal{P}$ in which y_2 is a strict descendant of y_1. Since b is a label of $\mathcal{T}_1$, this implies that y_2 is a strict descendant of b in $\mathcal{T}_1$, so, by definition, there is an arc in $D(\mathcal{P})$ from b to y_2. Repeating this argument for y_2 and y_3, we deduce that there is an arc in $D(\mathcal{P})$ from b to y_3. Continuing in this way, we eventually establish that there is an arc in $D(\mathcal{P})$ from b to x. A similar argument shows that there is an arc (x, b) in $D(\mathcal{P})$ if there is a directed path in $D(\mathcal{P})$ from x to b. This establishes (i).

We now prove (ii). Since (y, x) is an arc of $D(\mathcal{P})$, there is a tree $\mathcal{T}$ in $\mathcal{P}$ for which x is a strict descendant of y. But this means that, as (b, x) is an arc of $D(\mathcal{P})$, b is a common label and is pairwise consistent with x, and all trees in $\mathcal{P}$ are singularly labeled, either b is a strict descendant of y or y is a strict descendant of b in $\mathcal{T}$. In particular, either (b, y) or (y, b) is an arc in $D(\mathcal{P})$, respectively. □

PROPOSITION 3.6. *Let $\mathcal{P}$ be a collection of rooted semilabeled trees that are singularly labeled and let $\mathcal{P}'$ be a collection of fully labeled trees obtained from $\mathcal{P}$ by adding distinct new labels. If $\mathcal{P}$ is pairwise consistent and the descendancy graph of $\mathcal{P}'$ is acyclic, then* NESTEDSUPERTREE *applied to $\mathcal{P}$ returns a rooted semilabeled tree.*

Proof. Because of Lemma 3.4 and the fact that DESCENDANT successively considers proper restrictions of the modified descendancy graph of $\mathcal{P}'$, it suffices to show that one can always choose a nonempty set $\mathcal{S}_0$ of label vertices in steps 1 and 2 at each iteration of the subroutine DESCENDANT. To see this, suppose that at some iteration of DESCENDANT the associated connected restriction, D, say, of $D'(\mathcal{P}')$ has no label vertex of indegree zero. Let $\mathcal{S}$ be the set of label vertices of D in which the only incoming arcs are the ones coming from rooted triple vertices. Since any restriction of the descendancy graph of $\mathcal{P}'$ is acyclic, it follows that $\mathcal{S}$ is nonempty. Let a_1 be an element of $\mathcal{S}$. By the construction of the modified descendancy graph, a_1 is a label of a common rooted triple, $a_1a_2|b$, say, of $\mathcal{P}$. Furthermore, a_1 is not the only element of $\mathcal{S}$; for otherwise, every label vertex of D, including a_2, would be a strict descendant of a_1. It now follows that, as D is connected, there is a label vertex w that lies in a directed path from a_1 and that also lies in a directed path from a common label vertex, x_1, say, that is distinct from a_1 and is in $\mathcal{S}$. By Lemma 3.5(i), this implies that there exists a tree $\mathcal{T}_1$ in $\mathcal{P}$ in which w is a strict descendant of a_1 and a tree $\mathcal{T}_2$ in $\mathcal{P}$ in which w is a strict descendant of x_1. As a_1 and x_1 are not comparable in $\mathcal{T}_1$, it follows that w is not comparable to x_1 in $\mathcal{T}_1$. But w is a strict descendant of x_1 in $\mathcal{T}_2$, contradicting the assumption that $\mathcal{P}$ is pairwise consistent. We conclude that at steps 1 and 2 of each iteration of Descendant, we can always find an appropriate nonempty set of label vertices. □

Proposition 3.7. *Let $\mathcal{P}$ be a collection of rooted semilabeled trees that are singularly labeled. Then the running time of* NestedSupertree *applied to $\mathcal{P}$ is polynomial in $|\mathcal{L}(\mathcal{P})| \times |\mathcal{P}|$.*

Proof. Let $\mathcal{P}'$ be a collection of rooted fully labeled trees that is obtained from $\mathcal{P}$ by adding distinct new labels. Since the only possible unlabeled vertices of a rooted semilabeled tree are either the root vertex or a vertex of degree at least three, the number of such interior vertices is at most one less than the number of leaves. Therefore, to prove the proposition, it suffices to show that the running time of NestedSupertree is polynomial in $|\mathcal{L}(\mathcal{P}')| \times |\mathcal{P}|$.

It is clear that checking for pairwise consistency is polynomial time in $|\mathcal{L}(\mathcal{P}')| \times |\mathcal{P}|$. Furthermore, the construction of the descendancy graph of $\mathcal{P}'$ can be also be done in such a time. Now one can determine if a directed graph has no directed cycles by successively deleting vertices (and their incident arcs) that have either indegree or outdegree zero. If this process results in the empty graph, then the original graph has no directed cycles; otherwise it has a directed cycle. Since the size of $D(\mathcal{P}')$ is polynomial in the size of $\mathcal{L}(\mathcal{P}')$, determining whether $D(\mathcal{P}')$ has no directed cycles is polynomial in the size of $\mathcal{L}(\mathcal{P}')$.

The number of triples of $\mathcal{L}(\mathcal{P})$ is polynomial in $|\mathcal{L}(\mathcal{P})|$ and so finding the collection of common rooted triples of $\mathcal{P}$ is also polynomial in $|\mathcal{L}(\mathcal{P}')| \times |\mathcal{P}|$. It follows that the construction of the modified descendancy graph of $\mathcal{P}'$ is polynomial time in $|\mathcal{L}(\mathcal{P}')| \times |\mathcal{P}|$. Lastly, as stated in the fourth remark following Example 3.2, at each iteration of the subroutine Descendant there is always at least one vertex with indegree zero. Consequently, at each iteration, $\mathcal{S}_0$ is nonempty and so the mixed graph resulting from deleting the elements in $\mathcal{S}_0$ is a proper restriction of the mixed graph inputted at that particular iteration. Thus the number of such iterations is bounded by the size of $\mathcal{L}(\mathcal{P}')$. We deduce that the running time of NestedSupertree is polynomial in $|\mathcal{L}(\mathcal{P}')| \times |\mathcal{P}|$. □

4. Other properties of NestedSupertree.

The main purpose of this section is to establish the rooted semilabeled tree analogue of desirable property (ii) in the introduction for NestedSupertree.

A rooted semilabeled tree $\mathcal{T}$ is *binary* if $\mathcal{T}$ is singularly labeled and every vertex has degree at most three except for the root, which has degree at most two. The main result of this section is the following theorem.

Theorem 4.1. *Let $\mathcal{P}$ be a collection of semilabeled trees that are singularly labeled and let $\mathcal{T}$ be a rooted semilabeled binary tree that is ancestrally displayed by each tree in $\mathcal{P}$. Suppose that* NestedSupertree *applied to $\mathcal{P}$ returns a rooted semilabeled tree $\mathcal{T}'$. Then $\mathcal{T}'$ ancestrally displays $\mathcal{T}$.*

To prove Theorem 4.1, we first establish several results. The first result, Proposition 4.2, is well known (for example, see [13]).

Proposition 4.2. *Let $\mathcal{T}$ be a rooted phylogenetic X-tree. Let*

$$\mathcal{R}(\mathcal{T}) = \{\mathcal{T}|S : S \subseteq X,\ |S| = 3,\ \mathcal{T}|S \textit{ is a rooted triple}\}.$$

If $\mathcal{T}'$ is a rooted phylogenetic X'-tree, where $X \subseteq X'$, and $\mathcal{R}(\mathcal{T}) \subseteq \mathcal{R}(\mathcal{T}')$, then $\mathcal{T}'$ displays $\mathcal{T}$.

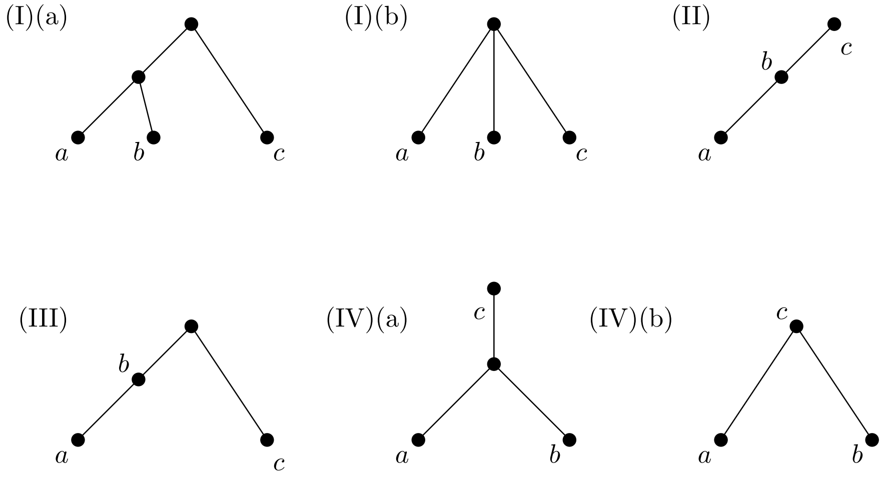


Fig. 5. *The six triples.*

For rooted semilabeled trees that are singularly labeled, the analogous result is Proposition 4.3. We will call a rooted semilabeled tree that is singularly labeled and has label set of size three a *triple*. A rooted triple is a particular type of triple. Up to isomorphism, there are six triples and these are shown in Figure 5. For convenience in this paper, we denote these triples as Types (I)(a) and (b), (II), (III), and (IV)(a) and (b). We will continue to refer to a triple of Type (I)(a) as a rooted triple.

Proposition 4.3. *Let $\mathcal{T}$ be a rooted semilabeled tree on X. Let*

$$\mathcal{B}(\mathcal{T}) = \{\mathcal{T}|S : S \subseteq X,\ |S| = 3,\ \mathcal{T}|S \textit{ is a triple of Type } \text{(I)(a)} \textit{ or } \text{(IV)(a)}\},$$

$$\mathcal{D}(\mathcal{T}) = \{c <_{\mathcal{T}} a : a, c \in \mathcal{L}(\mathcal{T})\},$$

and

$$\mathcal{N}(\mathcal{T}) = \{a \textit{ is not comparable to } b \textit{ under } \leq_{\mathcal{T}} : a, b \in \mathcal{L}(\mathcal{T})\}.$$

If $\mathcal{T}'$ is a rooted semilabeled tree on X', where $X \subseteq X'$, and $\mathcal{B}(\mathcal{T}) \subseteq \mathcal{B}(\mathcal{T}')$, $\mathcal{D}(\mathcal{T}) \subseteq \mathcal{D}(\mathcal{T}')$, and $\mathcal{N}(\mathcal{T}) \subseteq \mathcal{N}(\mathcal{T}')$, then $\mathcal{T}'$ ancestrally displays $\mathcal{T}$.

Proof. To prove the proposition, it is clear that we may assume that X and X' are the same sets. Let $\mathcal{T} = (T; \phi)$. The proof is by induction on the number n of interior labels of $\mathcal{T}$. If $n = 0$, then it is straightforward to deduce the result by Proposition 4.2 and the fact that $\mathcal{N}(\mathcal{T}) \subseteq \mathcal{N}(\mathcal{T}')$. Now assume that the result holds for all rooted semilabeled trees that have fewer than n interior labels, where $n \geq 1$. Since $\mathcal{T}$ has at least one interior label, there exists an interior vertex u of T that is labeled by an element, d, say, of X such that all elements of X that are strict descendants of d label leaves of $\mathcal{T}$. Let $\mathcal{T}_1$ be the rooted semilabeled tree obtained from $\mathcal{T}$ by replacing the rooted subtree of $\mathcal{T}$ that lies below or equal to u with a single leaf labeled by the elements of $\phi^{-1}(u)$. Let $\mathcal{T}_2$ be the rooted semilabeled tree that is the rooted subtree of $\mathcal{T}$ that lies below or equal to d and in which the elements in $\phi^{-1}(u)$ are removed. Note that if u has outdegree one, then u is deleted and the root of $\mathcal{T}_2$ is the vertex of $\mathcal{T}$ that is immediately below u.

Now consider $\mathcal{T}'$. Since $\mathcal{D}(\mathcal{T}) \subseteq \mathcal{D}(\mathcal{T}')$, each element in $\phi^{-1}(u)$ labels an interior vertex of $\mathcal{T}'$. Moreover, as there is an element of X that is a strict descendant of each element in $\phi^{-1}(u)$, it follows that, for all pairs $a, b \in \phi^{-1}(u)$, either a and b label the same vertex of $\mathcal{T}'$ or one element, a, say, is a strict descendant of b in $\mathcal{T}'$. Let c be a least element of $\phi^{-1}(u)$ under $\leq_{\mathcal{T}'}$ and let v be the interior vertex of $\mathcal{T}'$ that is labeled by c. Again, as $\mathcal{D}(\mathcal{T}) \subseteq \mathcal{D}(\mathcal{T}')$, the set of strict descendants of c in $\mathcal{T}'$ is exactly the label set of $\mathcal{T}_2$. Analogous to the constructions of $\mathcal{T}_1$ and $\mathcal{T}_2$ in the previous paragraph, construct $\mathcal{T}_1'$ and $\mathcal{T}_2'$ from $\mathcal{T}'$ using the vertex v instead of u. Evidently, $\mathcal{B}(\mathcal{T}_1) \subseteq \mathcal{B}(\mathcal{T}_1')$, $\mathcal{D}(\mathcal{T}_1) \subseteq \mathcal{D}(\mathcal{T}_1')$, and $\mathcal{N}(\mathcal{T}_1) \subseteq \mathcal{N}(\mathcal{T}_1')$, and $\mathcal{B}(\mathcal{T}_2) \subseteq \mathcal{B}(\mathcal{T}_2')$, $\mathcal{D}(\mathcal{T}_2) \subseteq \mathcal{D}(\mathcal{T}_2')$, and $\mathcal{N}(\mathcal{T}_2) \subseteq \mathcal{N}(\mathcal{T}_2')$. Furthermore, both $\mathcal{T}_1$ and $\mathcal{T}_2$ have fewer than n labeled interior vertices. Therefore, by our induction assumption, $\mathcal{T}_1'$ and $\mathcal{T}_2'$ ancestrally display $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively. By definition, it immediately follows that $\mathcal{T}'$ ancestrally displays $\mathcal{T}$ unless u has outdegree one and the vertex of $\mathcal{T}'$ labeled by c has outdegree at least two. But then, in this case, there are elements $a, b \in X$ such that $\mathcal{T}|\{a, b, c\}$ is of Type (IV)(a) and $\mathcal{T}'|\{a, b, c\}$ is of Type (IV)(b), contradicting the assumptions in the statement of the proposition. This completes the proof of Proposition 4.3. □

LEMMA 4.4. *Let $\mathcal{P}$ be a collection of rooted semilabeled trees that are singularly labeled and let $a, b \in \mathcal{L}(\mathcal{P})$. Suppose that* NESTEDSUPERTREE *applied to $\mathcal{P}$ returns a rooted semilabeled tree $\mathcal{T}$.*

(i) *If a is a strict descendant of b in some tree in $\mathcal{P}$, then a is a strict descendant of b in $\mathcal{T}$.*

(ii) *If $a, b \in \bigcap_{\mathcal{T} \in \mathcal{P}} \mathcal{L}(\mathcal{T})$, and a is not comparable to b in each tree in $\mathcal{P}$, then a is not comparable to b in $\mathcal{T}$.*

Proof. Part (i) immediately follows from the description of NESTEDSUPERTREE.

To prove (ii), let $\mathcal{P}'$ be the collection of rooted fully labeled trees that is obtained from $\mathcal{P}$ by adding distinct new labels in step 2 of NESTEDSUPERTREE. At some iteration of the running of the subroutine DESCENDANT, one of the label vertices a or b in some restriction of the modified descendancy graph $D'(\mathcal{P}')$ of $\mathcal{P}'$ has indegree zero. Consider the first such iteration and let D denote the corresponding connected mixed graph. Without loss of generality, we may assume that a has indegree zero in this restriction. To establish the lemma, it suffices to show by the construction of $\mathcal{T}$ that b is not a vertex of D.

Let V_a be the subset of vertices of D that are either label vertices lying on a directed path starting at a or rooted triple vertices where both adjacent label vertices lie on a directed path starting at a. Since a and b are not comparable in every tree in $\mathcal{P}$ and a has indegree zero, it follows by the contrapositive of Lemma 3.5(i) that $b \notin V_a$. Thus, to establish that b is not a vertex of D, it suffices to show that V_a is the vertex set of D. To see this, suppose that D contains an arc (z, x), where $z \notin V_a$, but $x \in V_a$. Clearly, x is a label vertex of D. Assume that z is also a label vertex of D. By Lemma 3.5(i), (a, x) is an arc of D. Therefore, as a has indegree zero, it follows by Lemma 3.5(ii) that there is an arc from a to z in D. This implies that $z \in V_a$, which is a contradiction. In fact, by extending this argument, it is easily seen that, ignoring rooted triple vertices, V_a contains all of the label vertices of D. It now follows by the definition of V_a that V_a is the vertex set of D. This completes the proof of the lemma. □

We remark here that the condition that a and b are common labels of $\mathcal{P}$ in the statement of Lemma 4.4(ii) cannot be weakened.

Let $\mathcal{P}$ be a collection of rooted semilabeled trees. A triple whose label set $\{a, b, c\}$

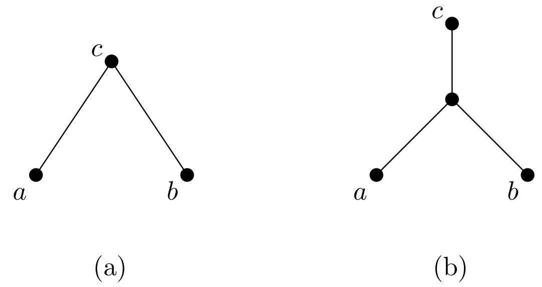


FIG. 6. *Two triples.*

is a subset of $\bigcap_{\mathcal{T}\in\mathcal{P}} \mathcal{L}(\mathcal{T})$ is *common* relative to $\mathcal{P}$ if, for all $\mathcal{T}_1, \mathcal{T}_2 \in \mathcal{P}$, $\mathcal{T}_1|\{a,b,c\}$ is isomorphic to $\mathcal{T}_2|\{a,b,c\}$.

LEMMA 4.5. *Let $\mathcal{P}$ be a collection of rooted semilabeled trees that are singularly labeled, and let $\mathcal{T}$ be a common triple of $\mathcal{P}$ of Type* (I)(a) *or* (IV)(a). *Let $\{a,b,c\}$ be the label set of $\mathcal{T}$. Suppose that* NESTEDSUPERTREE *applied to $\mathcal{P}$ returns a rooted semilabeled tree $\mathcal{T}'$. Then $\mathcal{T}'|\{a,b,c\}$ is isomorphic to $\mathcal{T}$.*

Proof. If $\mathcal{T}$ is a triple of Type (IV)(a), then it is easily seen, by interpreting Lemma 4.4 for a collection of rooted fully labeled trees that are singularly labeled, that $\mathcal{T}'|\{a,b,c\}$ is isomorphic to $\mathcal{T}$. Therefore suppose that $\mathcal{T}$ is the rooted triple $ab|c$, say. Let $\mathcal{P}'$ be the collection of rooted fully labeled trees that is obtained from $\mathcal{P}$ by adding distinct new labels in step 2 of NESTEDSUPERTREE. Since a, b, and c are common labels of $\mathcal{P}$, it follows by Lemma 4.4 that every pair of a, b, and c are not comparable in $\mathcal{T}$. Furthermore, by step 3 of DESCENDANT, a and b are always in the same arc component of the restrictions of the modified descendancy graph of $\mathcal{P}'$ that are considered throughout the running of NESTEDSUPERTREE provided c is in the same restriction. We now deduce from the description of DESCENDANT that $\mathcal{T}'|\{a,b,c\}$ is isomorphic to $ab|c$. □

We now prove Theorem 4.1.

Proof of Theorem 4.1. Since each label of $\mathcal{T}$ is a common label of $\mathcal{P}$, it immediately follows by Lemma 4.4 that $\mathcal{D}(\mathcal{T}) \subseteq \mathcal{D}(\mathcal{T}')$ and $\mathcal{N}(\mathcal{T}) \subseteq N(\mathcal{T}')$. Furthermore, by Lemma 4.5, $\mathcal{B}(\mathcal{T}) \subseteq \mathcal{B}(\mathcal{T}')$, and hence, by Proposition 4.3, $\mathcal{T}'$ ancestrally displays $\mathcal{T}$. □

COROLLARY 4.6. *Let $\mathcal{P}$ be a collection of rooted semilabeled trees that are singularly labeled. Suppose that* NESTEDSUPERTREE *applied to $\mathcal{P}$ returns a rooted semilabeled tree $\mathcal{T}'$. Then the following hold:*

(i) *If $\mathcal{T}$ is a common triple of $\mathcal{P}$, then $\mathcal{T}'$ ancestrally displays $\mathcal{T}$.*

(ii) *Let $\{a,b,c\}$ be a subset of $\bigcap_{\mathcal{T}\in\mathcal{P}} \mathcal{L}(\mathcal{T})$. Suppose that, for all $\mathcal{T} \in \mathcal{P}$, $\mathcal{T}|\{a,b,c\}$ is one of the two triples shown in Figure* 6. *Then $\mathcal{T}'$ ancestrally displays the triple shown in Figure* 6(b).

Proof. If $\mathcal{T}$ is a common triple of any type except Type (I)(b), then (i) follows from Theorem 4.1. If $\mathcal{T}$ is a common triple of Type (I)(b), then (i) follows from Lemma 4.4(ii).

For (ii), a routine check using both parts of Lemma 4.4 establishes this part of the corollary. □

We end this section with an observation regarding the last corollary. Observe that for the two triples in (ii) of this corollary one is a refinement of the other. Amongst the other triples only one other pair has this property, Types (I)(a) and (b). Despite part (ii) of Corollary 4.6, it is straightforward to construct an example where the analogue of (ii) for Types (I)(a) and (I)(b) does not hold. This is not a weakness of NESTEDSUPERTREE but simply highlights the fact shown in [14] that no general supertree method for rooted phylogenetic trees (and hence rooted semilabeled trees) is able to satisfy this analogue.

5. A variant of NESTEDSUPERTREE.

In this section, we present a particular variant of NESTEDSUPERTREE. This algorithm, which we call MINEDGEWEIGHTTREE, allows the input trees to be weighted and also satisfies the symmetry properties of ordering and renaming. To describe MINEDGEWEIGHTTREE, we simply note that it is obtained from NESTEDSUPERTREE by replacing step 2 of DESCENDANT with the following:

$2'$. If $\mathcal{S}_0$ is empty, then choose $\mathcal{S}_0$ as follows:
 (a) Let $\mathcal{C}_0$ denote the set of label vertices of $D'(\mathcal{P}')$ that have indegree zero.
 (b) For each $c \in \mathcal{C}_0$, weight c to be the sum of the weights of the trees in $\mathcal{P}'$ that induce at least one incident edge with c in $D'(\mathcal{P}')$.
 (c) Let $\mathcal{S}_0$ consist of the elements of $\mathcal{C}_0$ with minimum weight.

Note that if the input trees are not weighted, choose each tree to have weight one.

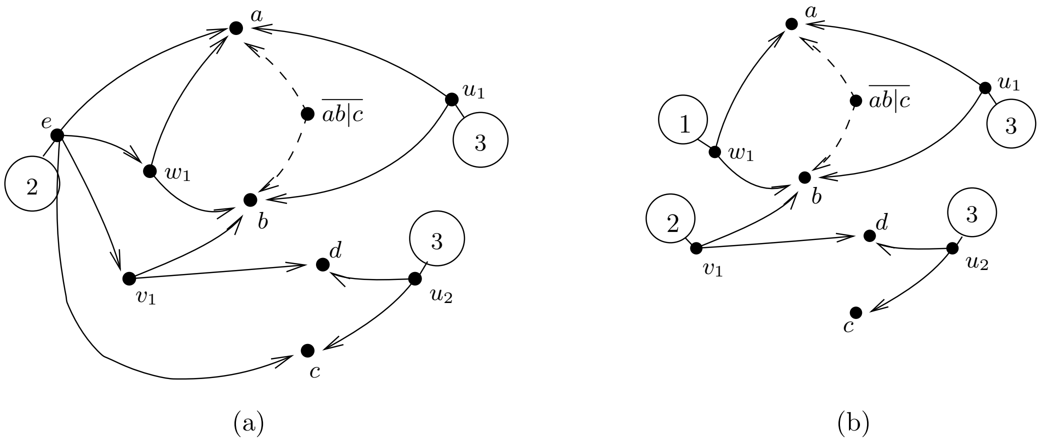
Remarks.

1. Clearly, at each iteration of the subroutine of MINEDGEWEIGHTTREE analogous to DESCENDANT, $\mathcal{S}_0$ is nonempty at the end of either step 1 or step $2'$. Furthermore, the time taken to find $\mathcal{S}_0$ is polynomial in $|\mathcal{L}(\mathcal{P})| \times |\mathcal{P}|$. It immediately follows by the results established in sections 3 and 4 for NESTEDSUPERTREE that MINEDGEWEIGHTTREE applied to a collection of rooted semilabeled trees that are singularly labeled and weighted satisfies the rooted semilabeled tree analogues of (i)–(iii) and (v) in the introduction.

2. In comparison with DESCENDANT, the set $\mathcal{S}_0$ of label vertices is well defined in the corresponding subroutine of MINEDGEWEIGHTTREE. Since no appeal is made to the specific symbols used as labels or to the order in which the members of $\mathcal{P}$ are listed in MINEDGEWEIGHTTREE, it follows that MINEDGEWEIGHTTREE also satisfies the rooted semilabeled tree analogues of (iv)(a) and (b) in the introduction.

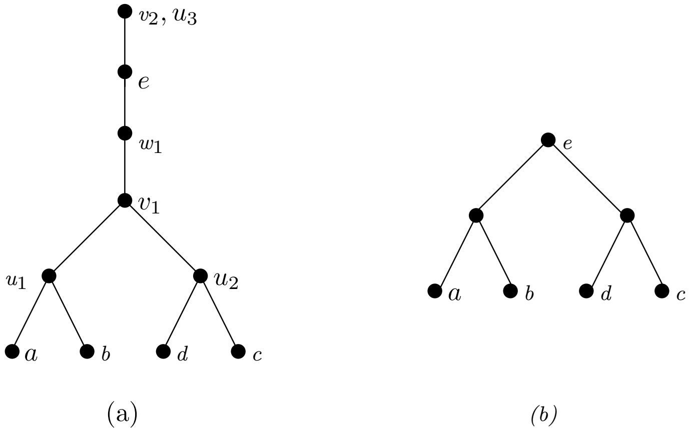
Example 5.1. To illustrate MINEDGEWEIGHTTREE, consider the collection $\mathcal{P}$ of rooted semilabeled trees described in Example 3.1 and the collection $\mathcal{P}'$ of rooted fully labeled trees obtained from $\mathcal{P}$ by adding distinct new labels. For the purposes of the example, suppose that the three trees in Figure 1 are weighted so that the leftmost tree is weighted 3, the middle tree is weighted 2, and the rightmost tree is weighted 1. The modified descendancy graph of $\mathcal{P}'$ is the same as that given in Figure 3.

Applying MINEDGEWEIGHTTREE to $\mathcal{P}$, the first iteration of its subroutine is the same as that in Example 3.2. In particular, $\mathcal{S}_0 = \{v_2, u_3\}$ at the end of step 1 and so, in this iteration, no label vertices of the inputted graph are weighted. In the second iteration of its subroutine, $\mathcal{S}_0$ is empty after step 1. At step $2'$(a), the set $\mathcal{C}_0$ of label vertices of the inputted graph with no incoming arcs is $\{e, u_1, u_2\}$. Since the label vertex e has exactly one incident edge and this is induced by the tree with weight 2, we give e weight 2 at step $2'$(b) in this iteration. Similarly, u_1 and u_2 are both weighted 3. This weighting together with the associated mixed graph are shown in Figure 7(a), where the edges and the arcs (c, a) in which a is not an immediate descendant of c are omitted. At step $2'$(c), $\mathcal{S}_0 = \{e\}$ and so, at this iteration, it is e and its incident arcs and edges that are deleted from the input graph. The graph resulting from these deletions is shown in Figure 7(b), where the weights of the label vertices with indegree zero are also shown. Continuing in this way, the subroutine



(a) (b)

FIG. 7. *The associated graphs in the second and third iteration of* DESCENDANTSUPERTREE *in Example* 5.1.



(a) *(b)*

FIG. 8. *The trees returned by* MINEDGEWEIGHTTREE *and its subroutine in Example* 5.1.

of MINEDGEWEIGHTTREE eventually returns the rooted fully labeled tree shown in Figure 8(a), and MINEDGEWEIGHTTREE returns the rooted semilabeled tree shown in Figure 8(b).

Remarks. Although we think MINEDGEWEIGHTTREE is a reasonable algorithm, we expect there to be more elaborate algorithms for supertree construction based on NESTEDSUPERTREE. The point is that it highlights how NESTEDSUPERTREE can be used as a basis for constructing new supertree methods for rooted semilabeled trees that satisfy all the rooted semilabeled tree analogues of the properties listed in the introduction.

6. NESTEDSUPERTREE **applied to rooted phylogenetic trees.** Although not originally intended for phylogenetics, the algorithm BUILD [2] was one of the first supertree methods for collections $\mathcal{P}$ of rooted phylogenetic trees. Furthermore, as well as MINCUTSUPERTREE and its modified version, the general approach taken by BUILD has been used in a number of more recent supertree algorithms, for example, [5, 6, 8, 11]. In the setting of phylogenetics, BUILD is a polynomial-time algorithm for deciding if $\mathcal{P}$ is compatible. In this section, we describe how NestedSupertree can be applied to $\mathcal{P}$ to determine the compatibility of $\mathcal{P}$. In the case that $\mathcal{P}$ is compatible, we also show that the rooted phylogenetic tree returned by NestedSupertree is the same as that returned by Build.

Since a collection $\mathcal{P}$ of rooted phylogenetic trees is compatible if and only if it is ancestrally compatible, it follows by the discussion preceding Proposition 3.3 that NestedSupertree can be suitably modified to determine the compatibility of $\mathcal{P}$. Theorem 6.1 shows that when applied to the same collection of compatible rooted phylogenetic trees, the supertrees returned by NestedSupertree with this modification and Build are identical up to isomorphism.

Before stating Theorem 6.1, we first give a description of Build. Let $\mathcal{P}$ be a collection of rooted phylogenetic trees and let $\mathcal{S}$ be a subset of $\mathcal{L}(\mathcal{P})$. Let $[\mathcal{P}, \mathcal{S}]$ be the graph that has vertex set S and has an edge joining two vertices a and b precisely if there exists a $c \in \mathcal{S}$ and a $\mathcal{T} \in \mathcal{P}$ such that

$$\mathcal{T}|\{a, b, c\} \cong ab|c.$$

```
Algorithm Build($\mathcal{P}, v$).
Input: A collection $\mathcal{P}$ of rooted phylogenetic trees.
Output: A rooted phylogenetic tree $\mathcal{T}$ that displays $\mathcal{P}$ with root vertex $v$, or the
statement $\mathcal{P}$ is not compatible.
1. Set $\mathcal{S}$ to be the label set of $\mathcal{P}$.
2. If $|\mathcal{S}| = 1$, then output the rooted phylogenetic tree consisting of the single vertex
   $v$ labeled by the element in $\mathcal{S}$.
3. If $|\mathcal{S}| \geq 2$, construct $[\mathcal{P}, S]$.
4. Let $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k$ denote the vertex sets of the components of $[\mathcal{P}, \mathcal{S}]$. If $k = 1$,
   then halt and return $\mathcal{P}$ is not compatible.
5. For each $i \in \{1, 2, \ldots, k\}$, call Build($\mathcal{P}_i, v_i$), where $\mathcal{P}_i$ is the collection of rooted
   phylogenetic trees obtained from $\mathcal{P}$ by restricting each tree in $\mathcal{P}$ to $\mathcal{S}_i$. If Build($\mathcal{P}_i, v_i$)
   returns a tree, then attach $\mathcal{T}_i$ to $v$ via the edge $\{v_i, v\}$.
```

Theorem 6.1. *Let $\mathcal{P}$ be a collection of rooted phylogenetic trees, and suppose that $\mathcal{P}$ is compatible. Then, up to isomorphism, the rooted phylogenetic trees returned by* NestedSupertree *with the above modifications and* Build *when applied to $\mathcal{P}$ are identical.*

Proof. We begin the proof with two observations. Let $\mathcal{S}$ denote the label set of $\mathcal{P}$, and let $\mathcal{P}'$ be a collection of rooted fully labeled trees that is obtained from $\mathcal{P}$ by adding distinct new labels. The first observation is that the vertex set of each component of the graph $[\mathcal{P}, \mathcal{S}]$ is a union of maximal proper clusters of the trees in $\mathcal{P}$. For the second observation, consider the descendancy graph of $\mathcal{P}'$, and let $\mathcal{S}_0$ denote the set of vertices of $D(\mathcal{P}')$ that have indegree zero and no incident edges. Then the vertex sets of each arc component of $D(\mathcal{P}')\backslash\mathcal{S}_0$ is also a union of maximal proper clusters of the trees in $\mathcal{P}'$. From these two observations, it is easily deduced, for all $a, b \in \mathcal{S}$, that a and b are in the same component of $[\mathcal{P}, \mathcal{S}]$ if and only if a and b are in the same arc component of $D(\mathcal{P}')\backslash\mathcal{S}_0$.

Let $\mathcal{S}_i$ be the vertex set of a component of $[\mathcal{P}, \mathcal{S}]$ and let $\mathcal{S}'_i$ be the vertex set of the arc component of $D(\mathcal{P}')\backslash\mathcal{S}_0$ that contains $\mathcal{S}_i$. Let $\mathcal{P}_i$ be the collection of rooted phylogenetic trees obtained from $\mathcal{P}$ by restricting each tree in $\mathcal{P}$ to $\mathcal{S}_i$, and let $\mathcal{P}'_i$ be the collection of rooted semilabeled trees obtained from $\mathcal{P}'$ by restricting each tree in $\mathcal{P}'$ to $\mathcal{S}'_i$. It is easily seen that all of the trees in $\mathcal{P}'_i$ are fully labeled. Now the equivalence at the end of the last paragraph implies that $\mathcal{P}'_i$ could have been obtained from $\mathcal{P}_i$ by adding distinct new labels. Furthermore, the arc component of $D(\mathcal{P}')$ containing the elements of $\mathcal{S}_i$ is equal to the descendancy graph of $D(\mathcal{P}'_i)$. Since $[\mathcal{P}, \mathcal{S}]$ contains at least two components, this implies that the maximal proper clusters of the trees returned by NestedSupertree with the appropriate modifications and Build when applied to $\mathcal{P}$ are the same. Repeatedly applying this argument to $\mathcal{P}_i$ for all i, we eventually deduce that the two rooted phylogenetic trees returned by NestedSupertree with the appropriate modifications and Build are identical. □

We end this section by remarking on what happens when NestedSupertree is applied to an arbitrary collection $\mathcal{P}$ of rooted phylogenetic trees. Let $\mathcal{P}'$ be a collection of rooted fully labeled trees that is obtained from $\mathcal{P}$ by adding distinct new labels. Since each of the trees in $\mathcal{P}$ are phylogenetic, $\mathcal{P}$ is pairwise consistent, and the descendancy graph of $\mathcal{P}'$ is acyclic. It is now easily seen from the description of Descendant that NestedSupertree applied to $\mathcal{P}$ returns a rooted semilabeled tree and that this tree is phylogenetic. It now follows by Propositions 3.7 and 3.3, and Theorem 4.1 that NestedSupertree is a general supertree method for rooted phylogenetic trees that satisfies the desirable properties (i)–(iii) in the introduction.

Acknowledgments. We thank the anonymous referees for their constructive and valuable comments.

REFERENCES

[1] E. N. Adams III, *N-trees as nestings: Complexity, similarity, and consensus*, J. Classification, 3 (1986), pp. 299–317.

[2] A. V. Aho, Y. Sagiv, T. G. Szymanski, and J. D. Ullman, *Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions*, SIAM J. Comput., 10 (1981), pp. 405–421.

[3] O. R. P. Bininda-Emonds, J. L. Gittleman, and M. A. Steel, *The (super)tree of life: Procedures, problems, and prospects*, Ann. Rev. Ecology System., 33 (2002), pp. 265–289.

[4] O. R. P. Bininda-Emonds, ed., *Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life*, Comput. Biol. Ser., Kluwer, Dordrecht, The Netherlands, 2004.

[5] D. Bryant, C. Semple, and M. Steel, *Supertree methods for ancestral divergence dates and other applications*, in Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life, O. Bininda-Emonds, ed., Comput. Biol. Ser., Kluwer, Dordrecht, The Netherlands, 2004, pp. 129–150.

[6] M. Constantinescu and D. Sankoff, *An efficient algorithm for supertrees*, J. Classification, 12 (1995), pp. 101–112.

[7] P. Daniel and C. Semple, *Supertree algorithms for nested taxa*, in Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life, O. Bininda-Emonds, ed., Comput. Biol. Ser., Kluwer, Dordrecht, The Netherlands, 2004, pp. 151–171.

[8] M. P. Ng and N. C. Wormald, *Reconstruction of rooted trees from subtrees*, Discrete Appl. Math., 69 (1996), pp. 19–31.

[9] R. D. M. Page, *Modified mincut supertrees*, in Proceedings of the Second International Workshop on Algorithms in Bioinformatics, R. Guig and D. Gusfield, eds., Springer, New York, 2002, pp. 537–552.

[10] R. D. M. Page, *Taxonomy, supertrees, and the Tree of Life*, in Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life, O. Bininda-Emonds, ed., Comput. Biol. Ser., Kluwer, Dordrecht, The Netherlands, 2004, pp. 247–265.

[11] C. Semple, *Reconstructing minimal rooted trees*, Discrete Appl. Math., 127 (2003), pp. 489–503.

[12] C. Semple and M. Steel, *A supertree method for rooted trees*, Discrete Appl. Math., 105 (2000), pp. 147–158.

[13] C. Semple and M. Steel, *Phylogenetics*, Oxford University Press, Oxford, UK, 2003.

[14] M. Steel, A. W. M. Dress, and S. Böcker, *Simple but fundamental limitations on supertree and consensus tree methods*, Syst. Biol., 49 (2000), pp. 363–368.

[15] S. Willson, *Private communication*, 2003.

SIAM J. Discrete Math.
Vol. 19, No. 2, pp. 481–488

POLYNOMIAL REPRESENTATIONS OF SYMMETRIC PARTIAL BOOLEAN FUNCTIONS*

MART DE GRAAF† AND PAUL VALIANT‡

Abstract. For Boolean polynomials in $\mathbb{Z}_p$ of sufficiently low degree we derive a relation expressing their values on one level set in terms of their values on another level set. We use this relation to derive linear upper and lower bounds, tight to within constant factor, on the degrees of various *approximate majority functions*, namely, functions that take the value 0 on one level set, the value 1 on a different level set, and arbitrary 0-1 values on other Boolean inputs. We show sublinear upper bounds in the case of moduli that are not prime powers.

Key words. Boolean function complexity, polynomial interpolation, lower bounds on degree, polynomial representation of Boolean functions, approximate majority function

AMS subject classifications. 68Q17, 68R05, 05E05, 94C10

DOI. 10.1137/S0895480103433562

1. Introduction.

Methods bounding the degree of polynomials that represent Boolean functions have been important tools in complexity theory. These techniques have been used to obtain several results that shed light on the complexity of Boolean functions. In particular, such polynomial degree lower bounds have consequences for the constant-depth circuit complexity of the associated Boolean functions.

We say that a polynomial represents a Boolean function if the polynomial is nonzero when the Boolean function is TRUE and zero when it is FALSE. The functions AND, OR, and Majority have been studied extensively in this framework and are examples of the more general class of threshold functions. Specifically, a threshold function is one which has value TRUE iff the number of nonzero inputs is at least a certain threshold. For AND, OR, and Majority, the respective thresholds are the number of inputs n, one, and $n/2$ respectively. Most of the work in this area concerns polynomials that represent these functions either exactly or at a large fraction of the points. Our results instead bound the degree of a large class of Boolean functions with values fixed at only a small subset of the domain. In particular, we study the *approximate majority function*, which is defined for fixed A, B with $A < B$ as any function that is TRUE if exactly B of the inputs are TRUE, and FALSE if exactly A are TRUE. Using properties of the binomial coefficients, we provide a linear lower bound on the degree of polynomials representing such approximate majority functions. For example, if for some prime $p, n = 4p^k, A = n/4$, and $B = 3n/4$, we prove a lower bound linear in n/p on the degree of a polynomial representing this approximate majority function over $\mathbb{Z}_p$. Our general linear lower bounds, however, hold only

*Received by the editors August 25, 2003; accepted for publication (in revised form) February 8, 2005; published electronically November 4, 2005.
http://www.siam.org/journals/sidma/19-2/43356.html

†This author's work was done while at CWI, Amsterdam, The Netherlands (mgdgraaf@cwi.nl). This author's research was partially supported by the EU fifth framework project QAIP, IST–1999–11234, and by grant 612.055.001 from the Netherlands Organization for Scientific Research (NWO).

‡Stanford University, Stanford, CA 94309. Current address: MIT Computer Science and Artificial Intelligence Laboratory, The Stata Center, Building 32, 32 Vassar Street, Cambridge, MA 02139 (pvaliant@mit.edu). This author's research was supported in part by the Department of Defense Multidisciplinary University Research Initiative (MURI) program administered by the Army Research Office under grant DAAD19-00-1-0177.

modulo powers of primes. For composite moduli with multiple prime factors, we prove sublinear upper bounds.

Degree lower bounds for Boolean polynomials were first used by Razborov [Raz87] and Smolensky [Smo87] in the context of proving lower bounds on the size of constant-depth Boolean circuits. These results inspired much work on the degree of threshold and other functions over various rings. Beigel [Bei93] gives an overview of much of the earlier work in this area. For example, Barrington, Straubing, and Thérien [BST90] proved linear upper bounds on the degree of a polynomial representing the OR function over $\mathbb{Z}_m$ and showed that they are tight for prime m. These upper bounds were improved by Barrington, Beigel, and Rudich [BBR92] to be sublinear for the case of composite m. In the case of majority, Tsai [Tsai96] proves a lower bound for all m of $n/2$ on the degree of the majority function over $\mathbb{Z}_m$. The approximate majority function with $A = n/4$ and $B = 3n/4$ arises naturally in the context of quantum complexity [GP01]. We show that the degree of this function is within a constant factor of that of the majority function for prime powers but significantly lower otherwise.

2. Preliminaries.

2.1. Combinatorics.

For natural numbers n and k, we denote by $(n)_k$ the k-ary representation of n, i.e., the string $\dots a_2 a_1 a_0$ with $0 \leq a_i < k$ such that $n = \sum_i a_i k^i$. Note that the first (from the right) nonzero digit of $(n)_k$ is given by the least i such that $k^{i+1} \nmid n$, an observation to which we shall frequently refer.

In 1878 Lucas [Luc78] gave a method for easily determining the value of $\binom{n}{k}$ mod p for prime p. This result is now known as Lucas's theorem. It is one of the main ingredients in the proofs of our results. By $x[i]$ we denote the symbol at the ith position from the right of string x.

Theorem 1 (see [Luc78]). *Let p be a prime number, and let n, k be positive integers. Then*

$$\binom{n}{k} \equiv \prod_{i=0}^{m} \binom{(n)_p[i]}{(k)_p[i]} \pmod{p},$$

where m is the maximal index i such that $(n)_p[i] \neq 0$ or $(k)_p[i] \neq 0$ and where we use the convention that $\binom{a}{x} = 0$ whenever $x > a$.

2.2. Representation of Boolean functions over $\mathbb{Z}_m$.

We now define what it means for a polynomial over $\mathbb{Z}_m$ to represent a Boolean function. We should note that there are several ways to represent a Boolean function by a polynomial over $\mathbb{Z}_m$, as discussed, for instance, in Tardos and Barrington [TB95]. The definition we use here is what is sometimes called *one-sided representation*.

Definition 1. *Let $g : \{0,1\}^n \to \{0,1\}$ be a Boolean function and $P : \mathbb{Z}_m^n \to \mathbb{Z}_m$ a multilinear polynomial. We say that P represents g over $\mathbb{Z}_m$ iff for all $x \in \{0,1\}^n$, $P(x) \equiv 0 \Leftrightarrow g(x) = 0$. By the degree* $\deg(P)$ *of a polynomial $P : \mathbb{Z}_m^n \to \mathbb{Z}_m$, we mean the degree of its largest monomial. The degree of a Boolean function $g : \{0,1\}^n \to \{0,1\}$ over $\mathbb{Z}_m$ is then defined as* $\deg(g, m) = \min\{\deg(P) \mid$ *P represents g over $\mathbb{Z}_m\}$.*

Note that since for all $x \in \{0,1\}$ and $\ell > 0$, we have that $x^\ell = x$, the restriction to *multilinear* polynomials is without loss of generality.

We will sometimes restrict ourselves to polynomials with outputs in $\{0,1\}$, which thus *strictly* represent Boolean functions. When the modulus is a prime power p^k, the following lemmas relate the degrees of the *strict* and *one-sided* representations to be within a factor of $(p-1)(2p^{k-1}-1)$. Both are usually stated as being folklore results. See [Bei93] for an overview of these and other similar results. The proof of Lemma 3 is due to Richard Beigel [Bei02], correcting a misstatement in [Bei93].

LEMMA 2. *Let p be a prime and $g : \mathbb{Z}_p^n \to \mathbb{Z}_p$ be a polynomial of degree d; then there is a polynomial $h : \mathbb{Z}_p^n \to \mathbb{Z}_p$ of degree $(p-1)d$ such that for all $x \in \{0,1\}^n$, $h(x) \in \{0,1\}$, and $h(x) \equiv 0$ iff $g(x) \equiv 0$.*

Proof. Take $h = g^{p-1}$. By Fermat's little theorem, $h(x) \equiv 1 \pmod{p}$ iff $g(x) \neq 0$. □

LEMMA 3. *Let k be a positive integer and p be a prime. If $g : \mathbb{Z}_{p^k}^n \to \mathbb{Z}_{p^k}$ is a polynomial of degree d, then there exists a degree $d(2p^{k-1}-1)$ polynomial $h : \mathbb{Z}_p^n \to \mathbb{Z}_p$, such that for all $x \in \{0,1\}^n$, $h(x) \equiv 0$ iff $g(x) \equiv 0$.*

Proof. By Theorem 1, we have that for every prime p and positive integer m,

$$\binom{m}{p^i} \equiv \binom{(n)_p[i]}{1} \equiv (n)_p[i] \pmod{p}.$$

Thus we have that for every such p, m

$$m \equiv 0 \pmod{p^k} \Leftrightarrow \forall i < k \left[\binom{m}{p^i} \equiv 0 \pmod{p} \right]. \tag{2.1}$$

Define the ith elementary symmetric function of the n variables $y_1, \ldots, y_n$, $i \leq n$, as

$$\sum_{1 \leq \ell_1 < \cdots < \ell_i \leq n} \prod_{j=1}^{i} y_{\ell_j}.$$

Note that if each $y_i \in \{0,1\}$, and exactly $|y|$ of them are 1, then the value of the above expression is $\binom{|y|}{i}$. Now write g as a sum of monomials of coefficient 1, i.e., replace, for example, $3x_1x_2$ by $x_1x_2 + x_1x_2 + x_1x_2$. Let $\binom{g(x)}{i}$ be the ith elementary symmetric function of the monomials in g. Define $h(x)$ as

$$h(x) = \sum_{i=0}^{k-1} \binom{g(x)}{p^i} \prod_{j=0}^{i-1} \left(1 - \binom{g(x)}{p^j}^{p-1} \right).$$

We have that the degree of $\binom{g(x)}{p^i}$ is $dp^i \leq dp^{k-1}$. Also, the degree of the product is at most $\sum_{j=0}^{k-2} d(p-1)p^j = d(p^{k-1}-1)$. Thus the degree of $h(x)$ is $d(2p^{k-1}-1)$. If $g(x) \equiv 0 \pmod{p^k}$, then by (2.1), $\binom{g(x)}{p^i} \equiv 0 \pmod{p}$ for all $0 \leq i < k$; hence $h(x) \equiv 0 \pmod{p}$. On the other hand, if $g(x) \not\equiv 0 \pmod{p^k}$, then using (2.1), let r be the least value such that $\binom{g(x)}{p^r} \not\equiv 0 \pmod{p}$. Note that the rth term in $h(x)$ is nonzero modulo p, but all the others are zero modulo p, since all terms after the rth contain the factor $(1 - \binom{g(x)}{p^r}^{p-1}) \equiv 0$, and hence $h(x) \not\equiv 0 \pmod{p}$. □

3. Level set relations. In this section we restrict ourselves to the field $\mathbb{Z}_p$, where p is a prime. For a binary string x, let $|x|$ denote its Hamming weight, the number of 1's. Note that in the following, we often identify an input $x \in \{0,1\}^n$ with the set $S = \{x_i \mid x_i = 1\}$. By definition $|x| = |S|$.

The following theorem relates the value of a polynomial at a set U with the sum of its values on subsets of U of a fixed cardinality, provided the polynomial is of sufficiently low degree.

THEOREM 4. *Let p be prime, and let $g : \mathbb{Z}_p^n \to \mathbb{Z}_p$ be a polynomial of degree at most p^r. Let $a < b$ be integers satisfying $\binom{b-1}{a-1} \not\equiv 0 \pmod{p}$. Then for any assignment $U \subset [n]$ with $|U| = bp^r$,*

$$g(U) \equiv (1 - b/a)g(\emptyset) + \binom{b-1}{a-1}^{-1} \sum_{\substack{|S|=ap^r \\ S\subset U}} g(S) \pmod{p}$$

unless $a \equiv 0 \pmod{p}$, in which case b/a is replaced by $\binom{b}{a}\binom{b-1}{a-1}^{-1}$.

Proof. Let $c_g(S)$ represent the coefficient in g of the term $\prod_{i\in S} x_i$. Then since g has degree at most p^r we can evaluate it at some point S with the following expression:

$$g(S) = \sum_{l\le p^r} \sum_{\substack{|Z|=l \\ Z\subset S}} c_g(Z).$$

Thus we have

$$\begin{aligned}\sum_{\substack{|S|=ap^r \\ S\subset U}} g(S) &= \sum_{\substack{|S|=ap^r \\ S\subset U}} \sum_{l\le p^r} \sum_{\substack{|Z|=l \\ Z\subset S}} c_g(Z) \\ &= \sum_{l\le p^r} \sum_{\substack{|S|=ap^r \\ S\subset U}} \sum_{\substack{|Z|=l \\ Z\subset S}} c_g(Z) \\ &= \sum_{l\le p^r} \sum_{\substack{|Z|=l \\ Z\subset U}} \binom{bp^r - l}{ap^r - l} c_g(Z),\end{aligned}$$

where the last equality holds since there are $\binom{bp^r-l}{ap^r-l}$ ways to choose the remaining $ap^r - l$ elements to form a set S with $Z \subset S \subset U$ of size ap^r.

From Lucas's theorem we have that for $0 < l \le p^r$, $\binom{bp^r-l}{ap^r-l} \equiv \binom{b-1}{a-1}$ and for $l = 0$, $\binom{bp^r}{ap^r} \equiv \binom{b}{a}$. Thus we may simplify the above as follows:

$$\begin{aligned}\sum_{l\le p^r} \sum_{\substack{|Z|=l \\ Z\subset U}} \binom{bp^r - l}{ap^r - l} c_g(Z) &\equiv \binom{b}{a} c_g(\emptyset) + \binom{b-1}{a-1} \sum_{l\le p^r} \sum_{\substack{|Z|=l \\ Z\subset U}} c_g(Z) \\ &= \left[\binom{b}{a} - \binom{b-1}{a-1}\right] g(\emptyset) + \binom{b-1}{a-1} \sum_{\substack{|Z|\le p^r \\ Z\subset U}} c_g(Z) \\ &= \left[\binom{b}{a} - \binom{b-1}{a-1}\right] g(\emptyset) + \binom{b-1}{a-1} g(U).\end{aligned}$$

Rearranging terms gives us the desired result. □

We would expect this theorem to be useful in proving degree lower bounds on polynomials representing Boolean functions whose values are specified only on certain level sets. We provide a few examples.

4. Lower bounds. As a first application, consider a Boolean function $g : \{0,1\}^n \to \{0,1\}$ that has $g(x) = 1$ if $|x| = n/4$ and $g(x) = 0$ if $|x| = 3n/4$, which can be thought of as the negation of an approximate majority function. We start with the special case when $n = 4p^k$ and prove that $\deg(g,p) = \Omega(n)$.

THEOREM 5. *Let p be a prime, $n = 4p^r$, and $g : \{0,1\}^n \to \{0,1\}$ be such that $g(x) = 0$ if $|x| = n/4$, and $g(x) = 1$ if $|x| = 3n/4$. Then*

$$\deg(g,p) > \frac{n}{4(p-1)}.$$

Proof. Consider any degree $d \leq \frac{n}{4(p-1)}$ multilinear polynomial P over $\mathbb{Z}_p$ that represents g. Using Lemma 2, transform P into a polynomial q that represents g over $\mathbb{Z}_p$ and that has $q(x) \in \{0,1\}$ for all $x \in \{0,1\}^n$. This will only increase the degree of q by a multiplicative factor $(p-1)$. We now prove a lower bound of $n/4$ on the degree of q.

Suppose for the sake of contradiction that we have such a polynomial of degree $n/4$. From Theorem 4 with $a = 1, b = 3, r = r$ we have

$$\begin{aligned} 1 \equiv g([3n/4]) &\equiv -2g(\emptyset) + \sum_{\substack{|S|=n/4 \\ S\subset[3n/4]}} g(S) \\ &= -2g(\emptyset) + 0. \end{aligned}$$

Thus for $g(\emptyset) \in \{0,1\}$ we have $2g(\emptyset) \equiv -1$, which implies that $p = 3$ and $g(\emptyset) = 1$. We now apply Theorem 4 again for $a = 1, b = 2, r = r$ to yield

$$\begin{aligned} g([2n/4]) &\equiv -1g(\emptyset) + \sum_{\substack{|S|=ap^r \\ S\subset[2n/4]}} g(S) \\ &= -1 + 0 \equiv 2, \end{aligned}$$

contradicting the fact that q is $0-1$ valued. Hence q must have degree greater than $n/4$. □

Using Lemma 3 we have the following corollary.

COROLLARY 6. *Let p be a prime, $n = 4p^r$ and let $g : \{0,1\}^n \to \{0,1\}$ be such that $g(x) = 0$ if $|x| = n/4$, and $g(x) = 1$ if $|x| = 3n/4$. Then*

$$\deg(g,p^k) > \frac{n}{4(2p^{k-1}-1)(p-1)}.$$

We note that in the above applications the number of variables n may be any integer $n \geq 3p^r$. We also note that the key to our proof is the fact that the degree of any polynomial *strictly* representing g is greater than $n/4$, which applies equally to the negation of g. Thus the preceding and following theorems apply equally to the approximate majority function as to its negation.

THEOREM 7. *Let p be a prime, $n \in \mathbb{Z}$, $A = ap^r, B = bp^r, A < B \leq n$ with neither b nor $\binom{b-1}{a-1}$ a multiple of p, and let $g : \{0,1\}^n \to \{0,1\}$ be such that $g(x) = 0$ if $|x| = A$, and $g(x) = 1$ if $|x| = B$. Then the degree of any polynomial over $\mathbb{Z}_p$ that* strictly *represents g is greater than p^r, with the following bound for the one-sided representation:*

$$\deg(g,p^k) > \frac{p^r}{(2p^{k-1}-1)(p-1)}.$$

Proof. As above we prove the degree bound for the strict representation and then apply Lemmas 2 and 3.

Suppose for the sake of contradiction there exists a polynomial P of degree $\leq p^r$ that strictly represents g over $\mathbb{Z}_p$. Note that the conditions of the theorem imply that $a \not\equiv 0$, for if $a \equiv 0$ and $b \not\equiv 0$, then Lucas's theorem would imply $\binom{b-1}{a-1} \equiv 0$, in violation of our assumptions. Thus from Theorem 4 we have that

$$1 \equiv (1 - b/a)g(\emptyset) + 0 \equiv (1 - b/a)g(\emptyset) \pmod{p}.$$

Since $g(\emptyset)$ is either 0 or 1, $g(\emptyset)$ must equal 1. Thus $b \equiv 0$, contradicting our assumption. Thus any *strictly* representing polynomial P must have degree greater than p^r, as desired. Note that the condition that $\binom{b-1}{a-1} \not\equiv 0 \pmod{p}$ is required by Theorem 4. □

5. Upper bounds. We now use Lucas's theorem to produce symmetric polynomials to represent approximate majority functions. In many cases, these polynomials have degrees relatively close to the lower bounds proved above.

We now work over the ring $\mathbb{Z}_m$, where m is some integer greater than 1. Given an approximate majority function $g(x)$ defined to be 0 when $|x| = A$ and 1 when $|x| = B$ for some A, B, we again wish to find a one-sided representing polynomial P such that $P \equiv 0 \pmod{m}$ iff $g = 0$. The strategy will be to find some number k such that

$$\binom{A}{k} \not\equiv \binom{B}{k} \pmod{m}$$

and then represent g as

$$P = \binom{x}{k} - \binom{A}{k}.$$

This leads to the following theorem.

THEOREM 8. *Given an approximate majority function $g : \{0,1\}^n \to \{0,1\}$ such that $g(x) = 0$ if $|x| = A$ and $g(x) = 1$ if $|x| = B$ for some $A, B \leq n$, then for $m > 1$, $\deg(g, m) \leq p^{r-1}$, where p^{r-1} is the smallest power of a prime factor of m such that $A \not\equiv B \pmod{p^r}$. Further, if m is squarefree, p^{r-1} is the minimum degree of a* symmetric *representing polynomial.*

Proof. Clearly if p^{r-1} is the smallest such power of a factor of m, then m contains exactly $r - 1$ factors of p. Thus the rth digits (from the right) in the base p representations of A and B must differ while the first $r - 1$ digits must be identical. From Lucas's theorem, these rth digits of A and B must equal $\binom{A}{p^{r-1}}$ and $\binom{B}{p^{r-1}}$, respectively, modulo p, which values must thus be different. Hence we may represent g as

$$P = \binom{x}{p^{r-1}} - \binom{A}{p^{r-1}},$$

where the notation $\binom{x}{p^{r-1}}$ is taken to mean the elementary symmetric polynomial on x of degree p^{r-1}. Clearly when $|x| = A$, $P(x) = 0$, and when $|x| = B$, $P(x) \not\equiv 0 \pmod{m}$ since $P(x) \not\equiv 0 \pmod{p}$.

Consider now the case where m is squarefree. Let k be the smallest degree of a symmetric function $\binom{x}{k}$ which differs on the levels A and B modulo m. Clearly any symmetric representing polynomial must have degree at least k, for otherwise it would have identical values on the levels A and B. We show $k \geq p^{r-1}$. Let q be some prime factor of m such that $\binom{A}{k} \not\equiv \binom{B}{k} \pmod{q}$. Then for some r' the r'th digits base q of A and B must differ. Consider the smallest such r'. Since $\binom{A}{k} \not\equiv \binom{B}{k} \pmod{q}$, Lucas's theorem implies $q^{r'-1} \leq k$. Since the r'th digits base q of A and B differ, we have $A \not\equiv B \pmod{q^{r'}}$. However, by hypothesis, p^{r-1} is the smallest power of a factor of m with this property, so $p^{r-1} \leq q^{r'-1}$. Thus $p^{r-1} \leq k$ as desired. ∎

We note that an alternate way of defining p^{r-1} is as follows. Factor $B - A$ as

$$B - A = p_1^{r_1} \dots p_j^{r_j}.$$

Then p^{r-1} as defined in Theorem 8 equals

$$\min_{p_i | m} p_i^{r_i}. \tag{5.1}$$

This leads to the following corollary.

Corollary 9. *Given an approximate majority function $g : \{0,1\}^n \to \{0,1\}$ such that $g(x) = 0$ if $|x| = A$ and $g(x) = 1$ if $|x| = B$ for some $A, B \leq n$, then for $m > 1$, $\deg(g, m) \leq (B - A)^{1/q}$, where q is the number of distinct prime factors of m.*

Proof. Factor $B - A$ as a product of powers of prime factors of m and some remaining factor. Clearly one of the q prime power factors must be at most $(B-A)^{1/q}$, implying the corollary by the above observation. ∎

We note that from (5.1), if a prime factor of m does not divide $B - A$, then the degree of the representing polynomial is 1!

Finally, we combine Theorems 7 and 8 to yield the following constant factor bound. (Note that if $p = 2$ the conditions of the theorem will never hold.)

Theorem 10. *Let p be a prime, $n \in \mathbb{Z}$, $A = ap^r, B = bp^r, A < B \leq n$ with neither $b - a$, b nor $\binom{b-1}{a-1}$ a multiple of p, and $g : \{0,1\}^n \to \{0,1\}$ such that $g(x) = 1$ if $|x| = A$, and $g(x) = 0$ if $|x| = B$. Then*

$$\frac{p^r}{(2p^{k-1} - 1)(p - 1)} < \deg(g, p^k) \leq p^r.$$

6. Discussion and open problems. We presented a relation between values of a low-degree polynomial on different level sets. We studied applications of this relation toward providing degree lower bounds for polynomials representing *approximate majority functions*. Further, many of these bounds lie surprisingly close to upper bounds given by symmetric functions. We note that an interesting consequence of the lower bound is a construction of an oracle separating EQP from $\mathsf{MOD_{p^k}P}$ [GP01] that is alternative to one implicit in [Bei91].

A number of open questions are left by this research. First, in $\mathbb{Z}_{p^k}$, Theorem 10 provides lower and upper bounds that differ by a factor of $(2p^{k-1} - 1)(p - 1)$. It would be interesting to see how this constant size gap can be closed. Theorem 10 relies on several conditions on the relation between A, B, and p, and we are curious to see which of these, if any, could be relaxed.

A possibly more fundamental open question raised by this paper is to find good lower bounds on the degree of approximate majority functions over $\mathbb{Z}_m$ for composite m. The techniques used in section 4 seem to break down here, even for squarefree m.

Acknowledgments. The first author would like to thank Harry Buhrman and Frederic Green for initially suggesting this problem and Leen Torenvliet and Ronald de Wolf for some interesting and useful discussions on the subject. The second author would like to thank Madhu Sudan for bringing this problem to his attention.

REFERENCES

[BBR92] D. Mix Barrington, R. Beigel, and S. Rudich, *Representing Boolean functions as polynomials modulo composite numbers*, in Proceedings of the 24th ACM Symposium on Theory of Computing, 1992, pp. 455–461.

[Bei91] R. Beigel, *Relativized counting classes: Relations among thresholds, parity, and mods*, J. Comput. System Sci., 42 (1991), pp. 76–96.

[Bei93] R. Beigel, *The polynomial method in circuit complexity*, in Proceedings of the 8th IEEE Structure in Complexity Theory Conference, 1993, pp. 82–95.

[Bei02] R. Beigel, *private communication*, October 2002.

[BST90] D. Mix Barrington, H. Straubing, and D. Thérien, *Non-uniform automata over groups*, Inform. and Comput., 89 (1990), pp. 109–132.

[GP01] F. Green and R. Pruim, *Relativized separation of EQP from P(NP)*, Inform. Process. Lett., 80 (2001), pp. 257–260.

[Luc78] E. Lucas, *Sur les congruences des nombres eulériens et les coefficients différentiels des fonctions trigonométriques, suivant un module premier*, Bull. Soc. Math. France, 6 (1878), pp. 49–54.

[Raz87] A. A. Razborov, *Lower bounds for the size of circuits of bounded depth with basis AND, OR*, Math. Notes Acad. Sci. USSR, 41 (1987), pp. 333–338.

[Smo87] R. Smolensky, *Algebraic methods in the theory of lower bounds for Boolean circuit complexity*, in Proceedings of the 19th Annual STOC, 1987, pp. 77–82.

[TB95] G. Tardos and D. Mix Barrington, *A lower bound on the mod* 6 *degree of the OR function*, in Israel Symposium on Theory of Computing Systems, 1995, pp. 52–56.

[Tsai96] S. C. Tsai, *Lower bounds on representing boolean functions as polynomials in* $\mathbb{Z}_m$, SIAM J. Discrete Math., 9 (1996), pp. 55–62.

SIAM J. DISCRETE MATH.
Vol. 19, No. 2, pp. 489–500

TWO-PART AND k-SPERNER FAMILIES: NEW PROOFS USING PERMUTATIONS*

PÉTER L. ERDŐS†, ZOLTÁN FÜREDI‡, AND GYULA O. H. KATONA†

Abstract. This is a paper about the beauty of the permutation method. New and shorter proofs are given for the theorem [P. L. Erdős and G. O. H. Katona, *J. Combin. Theory. Ser. A*, 43 (1986), pp. 58–69; S. Shahriari, *Discrete Math.*, 162 (1996), pp. 229–238] determining all extremal two-part Sperner families and for the uniqueness of k-Sperner families of maximum size [P. Erdős, *Bull. Amer. Math. Soc.*, 51 (1945), pp. 898–902].

Key words. extremal problems, Sperner families, permutation method

AMS subject classification. 05D05

DOI. 10.1137/S0895480104443588

1. Introduction.

Let X be a finite set of n elements. A family $\mathcal{F}$ of subsets of X is called *Sperner* (or *inclusion-free*, or an *antichain*) if $E, F \in \mathcal{F}$ implies $E \not\subset F$. The classic result of Sperner [15] states that

$$|\mathcal{F}| \leq \binom{n}{\lfloor \frac{n}{2} \rfloor} \tag{1}$$

with equality only when $\mathcal{F}$ consists either of all sets of size $\lfloor \frac{n}{2} \rfloor$ or of all sets of size $\lceil \frac{n}{2} \rceil$.

There are several generalizations and elegant proofs. However, frequently the case of equality is left to the reader, since it could be rather complicated. The aim for this paper is to illustrate the strength of the permutation method by presenting new shorter proofs for Sperner-type theorems. We will give two proofs, one using the permutation method and another using cyclic permutations, a method developed by the senior author [8], [9] and applied successfully to Sperner theorems by Füredi (see [10]).

1.1. Two-part families.

Kleitman [11] and Katona [7] independently observed that the statement of the Sperner theorem remains unchanged if the conditions are weakened in the following way. Let $X = X_1 \cup X_2$ be a partition of the underlying set X, $|X_i| = n_i$, $n_1 + n_2 = n$. Suppose $n_1 \geq n_2$ for the entire paper. We say that $\mathcal{F}$ is a *two-part Sperner family* if and only if $E, F \in \mathcal{F}$ $(E \neq F), E \subset F$ implies $(F - E) \not\subset X_1, X_2$. Kleitman [11] and Katona [7] proved that the size of a two-part Sperner family cannot exceed the right-hand side of (1).

The family of all $\lfloor \frac{n}{2} \rfloor$-element subsets gives equality here, too. There are, however, many other optimal constructions. A family $\mathcal{F}$ is called *homogeneous* (with respect

*Received by the editors May 9, 2004; accepted for publication (in revised form) December 31, 2004; published electronically November 4, 2005.
http://www.siam.org/journals/sidma/19-2/44358.html

†Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences, P.O. Box 127, Budapest 1364, Hungary (elp@renyi.hu, ohkatona@renyi.hu). The research of these authors was partially supported by Hungarian Foundation for Scientific Research grants (OTKA) T 037846 and T 034702.

‡Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences, P.O. Box 127, Budapest 1364, Hungary (furedi@renyi.hu). Current address: Department of Mathematics, University of Illinois at Urbana-Champaign, Urbana, IL 61801 (z-furedi@math.uiuc.edu). The research of this author was supported in part by Hungarian National Science Foundation grants OTKA T 032452 and T 037846 and National Science Foundation grant DMS 0140692.

to the partition X_1, X_2) if $F \in \mathcal{F}$ implies $E \in \mathcal{F}$ for all sets satisfying $|E \cap X_1| = |F \cap X_1|$, $|E \cap X_2| = |F \cap X_2|$. A homogeneous family can be described with the set $I(\mathcal{F}) = \{(i_1, i_2) : \ |F \cap X_1| = i_1, |F \cap X_2| = i_2 \text{ for some } F \in \mathcal{F}\}$. If $\mathcal{F}$ is a homogeneous two-part Sperner family, then $I(\mathcal{F})$ cannot contain pairs with the same first or second components, respectively. Consequently we have $|I(\mathcal{F})| \le n_2 + 1$. We say that a homogeneous family $\mathcal{F}$ is *full* if $|I(\mathcal{F})| = n_2 + 1$. Then for every i_2 $(0 \le i_2 \le n_2)$ there is a unique $f(i_2)$ such that $(f(i_2), i_2) \in I(\mathcal{F})$. A homogeneous family is called *well-paired* if it is full and

$$(2) \qquad \binom{n_2}{i} < \binom{n_2}{j} \text{ implies } \binom{n_1}{f(i)} \le \binom{n_1}{f(j)}$$

for every pair $1 \le i, j \le n_2$.

Here "well-paired" roughly means that every binomial coefficient of order n_2 obtains a match from the set of binomial coefficients of order n_1 and a larger value obtains a larger match. Of course this procedure is not unique. Let us illustrate the definition by an example. Let $n_1 = 8, n_2 = 5$. Since $(n_2 + 1 =)6$ largest binomial coefficients of order $n_1 = 8$ should be chosen, $\{f(0), f(1), f(2), f(3), f(4), f(5)\}$ is either $\{1, 2, 3, 4, 5, 6\}$ or $\{2, 3, 4, 5, 6, 7\}$. Choose the first case. $\binom{5}{2}$ and $\binom{5}{3}$ are the largest ones of the binomial coefficients of order 5; therefore $\binom{8}{f(2)}$ and $\binom{8}{f(3)}$ should be two largest ones from the binomial coefficients of order 8. Choose, for instance, $f(3) = 4, f(2) = 5$. Now $\binom{5}{1}$ and $\binom{5}{4}$ are larger than $\binom{5}{0}$ and $\binom{5}{5}$, so $\binom{8}{f(1)}$ and $\binom{8}{f(4)}$ should be next two largest ones after $\binom{8}{4}$ and $\binom{8}{5}$. Choose $f(4) = 3$ and $f(1) = 6$. Finally, let $f(0) = 1, f(5) = 2$. In this way we obtained a well-paired family $\mathcal{F}$ which consists of all subsets F satisfying $|F \cap X_1| = i_1$ and $|F \cap X_2| = i_2$, where $(i_1, i_2) \in \{(1, 0), (6, 1), (5, 2), (4, 3), (3, 4), (2, 5)\}$.

The following characterization (although not in this form) was proved in [5]. Later Shahriari [14] found an alternative proof.

THEOREM 1.1. *Let $\mathcal{F}$ be a two-part Sperner family with parts X_1, X_2, $|X_1| + |X_2| = n$. Then*

$$|\mathcal{F}| \le \binom{n}{\lfloor \frac{n}{2} \rfloor}$$

holds with equality if and only if $\mathcal{F}$ is a homogeneous well-paired family.

We give two new, probably shorter proofs in section 3 of the present paper.

Homogeneity type results are also true in a much more general setting. See the paper by Füredi et al. [6] or the joint paper of the present authors with Frankl [4]. In those papers it is shown, that there is a homogeneous optimal construction. Here we see that no other family can be optimal.

1.2. Families with no $k + 1$-chains.

To prove Theorem 1.1 we need another extension of the Sperner theorem, which is due to Paul Erdős. A family $\mathcal{F}$ of sets is called *k-Sperner* if it contains no *chain* $F_0 \subset F_1 \subset \cdots \subset F_k$ of $k + 1$ different sets. It was proved in [3] that if a family $\mathcal{F}$ of subsets of an n-element set is k-Sperner, then $|\mathcal{F}|$ is at most the sum of the k largest binomial coefficients of order n. The following theorem determines the cases of equality. This result is part of the folklore, but we do not know any written reference for it. The proof is a direct generalization of the uniqueness proof of the original Sperner theorem, due to the second author.

Theorem 1.2. *Let $\mathcal{F}$ be a k-Sperner family of subsets of an n-element set. Then*

$$|\mathcal{F}| \le \sum_{i=\lfloor(n-k+1)/2\rfloor}^{\lfloor(n+k-1)/2\rfloor} \binom{n}{i} \tag{3}$$

holds with equality if and only if $\mathcal{F}$ is the family of all sets of sizes either in the interval $[\lfloor\frac{(n-k+1)}{2}\rfloor, \lfloor\frac{(n+k-1)}{2}\rfloor]$ or in the interval $[\lceil\frac{(n-k+1)}{2}\rceil, \lceil\frac{(n+k-1)}{2}\rceil]$.

This theorem will be proved in section 2. The upper bound in the following result is an immediate corollary. Denote by $\binom{X}{i}$ the family of all i-element subsets of X; it is called the ith *level* in X.

Theorem 1.3. *Let $\mathcal{F} = \mathcal{F}_1 \cup \cdots \cup \mathcal{F}_k$ be a disjoint union of k-Sperner families of subsets of an n-element set X. Then $|\mathcal{F}|$ satisfies* (3) *with equality if and only if $\mathcal{F}_i = \binom{X}{r_i}$ holds for $1 \le i \le k$, where $r_1, \dots, r_k$ is a permutation of the elements either of the interval $[\lfloor\frac{(n-k+1)}{2}\rfloor, \lfloor\frac{(n+k-1)}{2}\rfloor]$ or of the interval $[\lceil\frac{(n-k+1)}{2}\rceil, \lceil\frac{(n+k-1)}{2}\rceil]$.*

2. Uniqueness in Erdős theorem and in the generalized YBLM-inequality. First we will prove a sharper version of Paul Erdős's theorem (Theorem 1.2) and will characterize the cases of equality of this sharper one. $\mathcal{F}$ is called *homogeneous* if $F \in \mathcal{F}, E \subset X$, and $|E| = |F|$ imply $E \in \mathcal{F}$. If $\mathcal{F}$ is a family of subsets, $f_i(\mathcal{F})$ will denote the number of i-element members of $\mathcal{F}$.

Theorem 2.1. *Let $\mathcal{F}$ be a k-Sperner family. Then*

$$\sum_{i=0}^{n} \frac{f_i(\mathcal{F})}{\binom{n}{i}} \le k \tag{4}$$

with equality only when $\mathcal{F}$ is homogeneous and contains sets of k distinct sizes.

The inequality part of this theorem can be found in [4, Theorem 5a] and is a generalization of the well-known YBLM-inequality [16], [1], [12], [13].

Proof. The method of cyclic permutations is used. The main point of this method is to reduce the original problem into an analogous problem on a fixed cyclic permutation.

If $\emptyset \in \mathcal{F}$, then $\mathcal{F} \setminus \{\emptyset\}$ is a $(k-1)$-Sperner family, and we can use induction on k. The case $X \in \mathcal{F}$ is similar. So from now on (in this section) we suppose that $f_0 = f_n = 0$ and $n > k$.

Let C be a cyclic permutation of X and let $\mathcal{F}(C)$ denote the subfamily of $\mathcal{F}$ consisting of all sets forming an interval (i.e., an arc) in C. $\mathcal{F}(C)$ is said to be *homogeneous* if $F \in \mathcal{F}(C)$ implies that every interval E along C of the same size ($|E| = |F|$) is in $\mathcal{F}(C)$. The proof is based on the following lemma.

Lemma 2.2.

$$|\mathcal{F}(C)| \le nk. \tag{5}$$

Here equality holds if and only if $\mathcal{F}(C)$ is homogeneous and it contains k distinct sizes.

Proof of Lemma 2.2. Since $\emptyset, X \notin \mathcal{F}$ at most k sets may start at any fixed element of X along C in one direction. This establishes (5).

In the case of equality there must be exactly k intervals in $\mathcal{F}(C)$ starting from each point of C. Let $B_i(j)$ $(1 \le i \le n,\ 1 \le j \le k)$ denote the jth interval starting from the ith point where $|B_i(1)| < |B_i(2)| < \cdots < |B_i(k)|$ is supposed. We claim that $|B_i(j)| \le |B_{i+1}(j)|$ holds. Indeed, otherwise $B_{i+1}(1) \subset B_{i+1}(2) \subset \cdots \subset B_{i+1}(j) \subset$

$B_i(j) \subset \cdots \subset B_i(k)$ would be a chain of intervals of length $k+1$, a contradiction. Hence we have $|B_1(j)| \le |B_2(j)| \le \cdots \le |B_n(j)| \le |B_1(j)|$ implying $|B_i(j)| = |B_{i+1}(j)|$ for all $1 \le i < n$ and $1 \le j \le k$. □

Let us return to the proof of Theorem 2.1. Lemma 2.2 yields

$$\sum_{C} \sum_{F \in \mathcal{F}(C)} 1 = \sum_{C} |\mathcal{F}(C)| \le (n-1)!nk = n!k. \tag{6}$$

The number of cyclic permutations C containing a given set F as an interval is $|F|!(n-|F|)!$ (if $|F| \ne 0, n$). Hence

$$\sum_{F \in \mathcal{F}} \sum_{C : F \in \mathcal{F}(C)} 1 = \sum_{F \in \mathcal{F}} |F|!(n-|F|)! \tag{7}$$

holds. Comparing (7) and (6) we obtain (4), the inequality part of Theorem 2.1.

Formula (4) can hold with equality only when (7) and (6) are equal, that is, when (5) holds with equality for all cyclic permutations: $\mathcal{F}(C)$ is homogeneous for each C. Consider any two subsets A and B ($\subset X$) of equal cardinality. It is obvious that there is a cyclic permutation C in which they are both intervals. Therefore either $A, B \in \mathcal{F}$ or $A, B \notin \mathcal{F}$ holds, and consequently $\mathcal{F}$ is also homogeneous. □

We need a simple inequality; for completeness we supply a sketch of the proof, standard in linear programming.

LEMMA 2.3. *Suppose that for integers $n \ge k \ge 1$ and nonnegative reals $f_1, f_2 \dots, f_{n-1}$ the following inequalities hold:*

$$\sum_{1 \le i \le n-1} \frac{f_i}{\binom{n}{i}} \le k,$$

$$f_i \le \binom{n}{i}.$$

Then

$$\sum_{1 \le i \le n-1} f_i \le \sum_{i=\lfloor (n-k+1)/2 \rfloor}^{\lfloor (n+k-1)/2 \rfloor} \binom{n}{i} := f(n,k).$$

Here equality holds if and only if

(a) *in the case $n \not\equiv k$ (mod 2), $f_i = \binom{n}{i}$ for $(n-k+1)/2 \le i \le (n+k-1)/2$ and $f_i = 0$ otherwise,*

(b) *in the case $n \equiv k$ (mod 2), $f_i = \binom{n}{i}$ for $(n-k+2)/2 \le i \le (n+k-2)/2$ and $f_{(n-k)/2} + f_{(n+k)/2} = \binom{n}{(n-k)/2}$ and $f_i = 0$ otherwise.*

Proof. Consider a vector $\mathbf{f} = (f_1, f_2, \dots, f_{n-1})$ which maximizes $\sum f_i$. (The domain is compact; maximum(s) exists.) For $\binom{n}{j} < \binom{n}{i}$ the inequalities $f_i < \binom{n}{i}, 0 < f_j$ lead to a contradiction, since replacing them by $f_i + \varepsilon\binom{n}{i}$ and $f_j - \varepsilon\binom{n}{j}$ keeps the constraint the lemma but increases the sum $\sum f_i$. □

Proof of Theorem 1.2. The constraint of Lemma 2.3 holds for the sequence $f_1(\mathcal{F}), \dots, f_{n-1}(\mathcal{F})$ by (4) and since $f_i(\mathcal{F}) \le \binom{n}{i}$ is obvious. This implies the Erdős theorem.

We can have equality in this theorem only when (4) holds with equality. Then Theorem 2.1 implies that $\mathcal{F}$ is homogeneous and consists of k distinct sizes. □

Proof of Theorem 1.3. The inequality part is trivial, since $\mathcal{F}$ is a k-Sperner family. It is clear from the previous proof that the equality implies equality in (4). Since $\mathcal{F}_i$ $(1 \le i \le k)$ is a Sperner family, (4) holds for $\mathcal{F}_i$ with $k = 1$. Hence (4) with $k = 1$ must hold with equality for each $\mathcal{F}_i$. Therefore $\mathcal{F}_i = \binom{X}{r_i}$ for some r_i. Since $\mathcal{F}_i$ are disjoint, r_i must be different, $\mathcal{F}$ is a union of k distinct levels. The maximality of $|\mathcal{F}|$ implies that these k levels must be the k middle ones. □

2.1. Uniqueness in the Erdős theorem using intervals. Here we give another proof for Theorem 1.2.

Let $\mathcal{F}$ be a k-Sperner family on the n-element underlying set $X = [n]$. We may suppose that $\emptyset, X \notin \mathcal{F}$ because these cases can easily be reduced to the general case. As in the classical proofs, consider a *permutation* π of X. The initial segments of π, i.e., the sets of the form $\{\pi(1), \pi(2), \ldots, \pi(i)\}_{1 \le i < n}$ form a *chain* $\mathcal{C}(\pi)$ of length $n-1$. The k-Sperner property of $\mathcal{F}$ implies that $\mathcal{C}(\pi)$ contains at most k members of $\mathcal{F}$, so we have

$$(8) \qquad \sum_{F: F\in\mathcal{F}, F\in\mathcal{C}(\pi)} \binom{n}{|F|} \le \sum k \,\text{largest binomial coefficients} := f(n,k).$$

Add this up for all the $n!$ permutations.

$$\sum_{\pi} \sum_{F\in\mathcal{F}, F\in\mathcal{C}(\pi)} \binom{n}{|F|} \le n! f(n,k).$$

Here the left-hand side can be determined exactly.

$$\sum_{F: F\in\mathcal{F}} \sum_{\pi: F\in\mathcal{C}(\pi)} \binom{n}{|F|} = \sum_{F} |F|!(n-|F|)! \binom{n}{|F|} = n! |\mathcal{F}|.$$

This gives $|\mathcal{F}| \le f(n,k)$.

If $|\mathcal{F}| = f(n,k)$, then equality holds in (8) for every π, so the sizes of the members of $\mathcal{F}$ in $\mathcal{C}(\pi)$ form a middle interval of length k. In the case $n \not\equiv k \pmod 2$ this middle interval is unique; we get that $\mathcal{F}$ is homogeneous, and it consists of all sets of sizes at least $(n-k+1)/2$ and at most $(n+k-1)/2$. In the case $n \not\equiv k \pmod 2$ there are two possibilities for a middle interval, so $f_i = \binom{n}{i}$ for $(n-k+2)/2 \le i \le (n+k-2)/2$ and $f_{(n-k)/2} + f_{(n+k)/2} = \binom{n}{(n-k)/2}$ and $f_i = 0$ otherwise. We also obtain that for $|F'| = (n-k)/2$, $|F''| = (n+k)/2$, $F' \subset F''$, one and only one of $\{F', F''\}$ belongs to $\mathcal{F}$. Suppose that there exists an $F \in \mathcal{F}$, $|F| = (n-k)/2$. We claim that $f_{(n-k)/2} = \binom{n}{(n-k)/2}$ and then $f_{(n+k)/2} = 0$, and we are done.

Consider an arbitrary pair $x \in F$ and $y \in X \setminus F$. We claim that $F \setminus \{x\} \cup \{y\} \in \mathcal{F}$. Indeed, consider a permutation π where $F \setminus \{x\}$, F and $F \cup \{y\}$ are initial segments, and let π' be a permutation obtained from π be exchanging the places of x and y. The largest member of $\mathcal{F}$ in $\mathcal{C}(\pi)$ has $(n+k-2)/2$ elements, so the same is true for $\mathcal{C}(\pi')$. Since the sizes of the members of $\mathcal{C}(\pi') \cap \mathcal{F}$ form a middle interval, the smallest member has $(n-k)/2$ elements. This smallest member is $F \setminus \{x\} \cup \{y\}$.

Call two $(n-k)/2$-element sets F_1 and F_2 neighbors if $|F_1 \cap F_2| = |F_1| - 1$. Then the above property of the extremal $\mathcal{F}$ can be formulated as it contains all neighbors of F whenever $F \in \mathcal{F}$. It follows that in that case it contains the second, third, etc. neighbors, so $\mathcal{F}$ contains the whole $((n-k)/2)$th level. □

3. Two-part Sperner families. In the method of cyclic permutations a given problem on subsets is reduced to intervals in a cyclic permutation of the underlying set. In the present proof the problem will be reduced to a family of certain mixed objects, pairs (A, B), where A is a subset of X_1 and B is an interval along a fixed cyclic permutation of X_2. Therefore the method can be called the *mixcyc* method.

First proof of Theorem 1.1. Let C_2 be a cyclic permutation of X_2 and $\mathcal{F}$ a family of subsets of X. Then $\mathcal{F}(C_2)$ will denote those members of $\mathcal{F}$ for which $F \cap X_2$ is an interval along C_2.

Introduce the notation

$$t(j) = \begin{cases} n_2 & \text{if} \quad j = 0, n_2, \\ 1 & \text{if} \quad 1 \le j \le n_2 - 1. \end{cases}$$

The double sum

$$\sum_{\substack{(C_2,F)\\ F\in\mathcal{F}(C_2)}} t(|F\cap X_2|)\binom{n_2}{|F\cap X_2|} \tag{9}$$

will be evaluated in two different ways. First

$$\sum_{F\in\mathcal{F}} \sum_{C_2\colon\, F\in\mathcal{F}(C_2)} t(|F\cap X_2|)\binom{n_2}{|F\cap X_2|}$$
$$= \sum_{F\in\mathcal{F}} t(|F\cap X_2|)\binom{n_2}{|F\cap X_2|} \sum_{C_2\colon\, F\in\mathcal{F}(C_2)} 1.$$

Here

$$\sum_{C_2\colon\, F\in\mathcal{F}(C_2)} 1 = \begin{cases} (n_2-1)! & \text{if } F\cap X_2 = \emptyset \text{ or } X_2, \\ |F\cap X_2|!\,(n-|F\cap X_2|)! & \text{otherwise.} \end{cases}$$

Therefore

$$(9) = \sum_{F\in\mathcal{F}} n_2! = |\mathcal{F}|n_2!.$$

On the other hand, (9) is equal to

$$\sum_{C_2} \sum_{F\in\mathcal{F}(C_2)} t\left(|F\cap X_2|\right)\binom{n_2}{|F\cap X_2|}. \tag{10}$$

Introduce the notation

$$w(i) = t(i)\binom{n_2}{i}, \qquad i = 0, \dots, n_2,$$

and let $(j_0, j_1, \dots, j_{n_2})$ be one of the permutations of $(0, 1, \dots, n_2)$ satisfying $w(j_0) \ge w(j_1) \ge \cdots \ge w(j_{n_2}) = n_2$. There are four cases of w with value n_2. Suppose that j_{n_2-1} and j_{n_2} are chosen to be 0 and n_2, respectively. Now fix a cyclic permutation $C_2 = (c_1, \dots, c_n)$ of X_2 and decompose its intervals into n_2 *chains* of intervals: define

$$\mathcal{L}_1 = \{\emptyset, \{c_1\}, \{c_1, c_2\}, \dots, \{c_1, c_2, \dots, c_{n_2-1}\}, \{c_1, \dots, c_{n_2}\}\},$$

while for $i = 2, \ldots, n_2$ let

$$\mathcal{L}_i = \{\{c_i\}, \{c_i, c_{i+1}\}, \ldots, \{c_i, c_{i+1}, \ldots, c_{n_2}, c_1, \ldots, c_{i-3}\}, \{c_i, \ldots, c_{i-2}\}\}\,.$$

Consider the subsum

$$\sum_{(F\cap X_2)\in\mathcal{L}_1} t(|F\cap X_2|)\binom{n_2}{|F\cap X_2|} = \sum_{i=0}^{n_2} |\mathcal{F}(j_i)|\, w(j_i), \tag{11}$$

where $\mathcal{F}(j)$ is defined by

$$\mathcal{F}(j) = \{F\cap X_1 : F\in\mathcal{F}, |F\cap X_2| = j \text{ and } F\cap X_2\in\mathcal{L}_1\}\,.$$

It is easy to see that the family $\mathcal{F}(j)$ is Sperner for every j and that $\mathcal{F}(j_k)\cap\mathcal{F}(j_l) = \emptyset$ holds when $k \neq l$. Formula (11) can be written as

$$\begin{aligned}(11) = {}& \Big(|\mathcal{F}(j_0)| + \cdots + |\mathcal{F}(j_{n_2})|\Big)w(j_{n_2})\\ &+ \Big(|\mathcal{F}(j_0)| + \cdots + |\mathcal{F}(j_{n_2-1})|\Big)\Big(w(j_{n_2-1}) - w(j_{n_2})\Big)\\ &+ \cdots + \Big(|\mathcal{F}(j_0)| + |\mathcal{F}(j_1)|\Big)\Big(w(j_1) - w(j_2)\Big)\\ &+ |\mathcal{F}(j_0)|\Big(w(j_0) - w(j_1)\Big).\end{aligned} \tag{12}$$

By the Erdős theorem the total size of k pairwise disjoint Sperner families in X_1 cannot exceed the k largest levels. Therefore if $m(i) = \binom{n_1}{i}$ and $(l_0, l_1, \ldots, l_{n_1})$ is one of the permutations of $(0, 1, \ldots, n_1)$ satisfying $m(l_0) \geq m(l_1) \geq \cdots \geq m(l_{n_1})$, then

$$\begin{aligned}(12) \leq {}& \Big(m(l_0) + m(l_1) + \cdots + m(l_{n_2})\Big)w(j_{n_2})\\ &+ \Big(m(l_0) + m(l_1) + \cdots + m(l_{n_2-1})\Big)\Big(w(j_{n_2-1}) - w(j_{n_2})\Big) + \cdots\\ &+ \Big(m(l_0) + m(l_1)\Big)\Big(w(j_1) - w(j_2)\Big) + m(l_0)\Big(w(j_0) - w(j_1)\Big)\\ = {}& \sum_{i=0}^{n_2} m(l_i)w(j_i).\end{aligned} \tag{13}$$

The same estimations can be applied for the other $n_2 - 1$ chains $\mathcal{L}_k$ $(k = 2, \ldots, n_2)$:

$$\sum_{F\cap X_2\in\mathcal{L}_k} t(|F\cap X_2|)\binom{n_2}{|F\cap X_2|} \leq \sum_{i=0}^{n_2-2} m(l_i)w(j_i).$$

Using the fact that the number of cyclic permutations C_2 is $(n_2 - 1)!$ and putting together the previous inequalities, we obtain

$$\begin{aligned}(10) &\leq \sum_{C_2}\left(n_2\sum_{i=0}^{n_2-2} m(l_i)w(j_i) + m(l_{n_2-1})w(j_{n_2-1}) + m(l_{n_2})w(j_{n_2})\right)\\ &= n_2!\sum_{i=0}^{n_2}\binom{n_1}{l_i}\binom{n_2}{j_i} = n_2!\sum_{i=0}^{n_2}\binom{n_1}{\lceil\frac{n_1+n_2}{2}\rceil + i}\binom{n_2}{i}\\ &= n_2!\sum_{i=0}^{n_2}\binom{n_1}{\lfloor\frac{n_1+n_2}{2}\rfloor - i}\binom{n_2}{i} = \binom{n}{\lfloor\frac{n}{2}\rfloor}.\end{aligned} \tag{14}$$

(9) = (10) ≤ (14) finishes the proof of the two-part Sperner theorem.

To prove the equality part of Theorem 1.1 we only have to check carefully the cases of equality in the above proof of the two-part Sperner theorem.

Define

$$\mathcal{F}_1(B) = \{A : \ A \subset X_1, A \cup B \in \mathcal{F}\} \quad \text{for } B \subset X_2.$$

If $\mathcal{F}$ is a family satisfying equality in the Erdős theorem (in the form of Theorem 1.3), then there must be equality between (12) and (13), that is,

$$|\mathcal{F}(j_0)| + |\mathcal{F}(j_1)| + \cdots + |\mathcal{F}(j_r)| = m(l_0) + m(l_1) + \cdots + m(l_r) \tag{15}$$

holds whenever $w(j_r) - w(j_{r+1}) > 0$ (where $w(j_{n_2+1}) = 0$). It is obvious that every second of these differences is zero, and the other ones are positive. If n_2 is even, then $w(j_0) - w(j_1)$ is positive, $w(j_1) - w(j_2)$ is zero, $w(j_2) - w(j_3)$ is positive, and so on. On the other hand, if n_2 is odd, then this sequence starts with a zero. We should not forget, however, that there are some irregularities at the end. First, the last coefficient $w(j_{n_2})$ (first in (12)) is always positive; second, it is preceded by three zeros. This implies, by Theorem 1.3, that in the case of even $n_2, \mathcal{F}(j_0)$ must be one of the (one or two) largest levels in X_1; $\mathcal{F}(j_0), \mathcal{F}(j_1), \mathcal{F}(j_2)$ must be the three largest levels; and so on. Hence $\mathcal{F}(j_1)$ and $\mathcal{F}(j_2)$ are the two levels next or equal in size. The same holds for $\mathcal{F}(j_{2s+1})$ and $\mathcal{F}(j_{2s+2})$ for $0 \le s \le \frac{n_2-6}{2}$. If n_2 is odd, then $\mathcal{F}(j_0)$ and $\mathcal{F}(j_1)$ are the two largest levels, $\mathcal{F}(j_2)$ and $\mathcal{F}(j_3)$ are the next two levels, and so on. In general $\mathcal{F}(j_{2s})$ and $\mathcal{F}(j_{2s+1})$ $(0 \le s \le \frac{n_2-5}{2})$ are a pair of the $(2s+1)$st and $(2s+2)$th largest levels.

Since $w(j_{n_2}) > 0$ holds, $\mathcal{F}(j_0), \dots, \mathcal{F}(j_{n_2})$ are the $n_2 + 1$ largest levels in X_1. However, we have some freedom in choosing their order, but this order must satisfy the conditions above. Until now we have proved a restricted version of the homogeneity of $\mathcal{F}$, namely, that the subfamily $\{F : \ F \in \mathcal{F}, F \cap X_2 \in \mathcal{L}_1\}$ is a homogenous full family. That is, the family $\{F \cap X_1 : \ F \in \mathcal{F}, F \cap X_2 = \{c_1, \dots, c_j\}\} = \mathcal{F}_1(\{c_1, \dots, c_j\}) = \mathcal{F}(j)$ is equal to $\binom{X_1}{w}$ for some w. Let this w be denoted by $f^*(j)$. It remained to check that this restriction of $\mathcal{F}$ is well-paired; that is, this ordering satisfies (2).

If n_2 is even, then the left-hand side of (2),

$$\binom{n_2}{j_u} < \binom{n_2}{j_v} \ (u < n_2 - 3), \tag{16}$$

holds if and only if $v \le u$ and u is not an even integer $= v + 1$. Then

$$\binom{n_1}{f^*(j_u)} \le \binom{n_1}{f^*(j_v)} \tag{17}$$

is obvious. The case when n_2 is odd is analogous. That is, the order follows (2) up to $n_2 - 4$. Consider now the case when $u = n_2 - 3, n_2 - 2, n_2 - 1, n_2$ and $n_2 - 3 > v$. Since $\{j_{n_2}, j_{n_2-1}\} = \{0, n_2\}$ by definition, consequently we have $\{j_{n_2-2}, j_{n_2-3}\} = \{1, n_2 - 1\}$, and hence the last few $\binom{n_2}{j_u}$ are $n_2, n_2, 1, 1$. (16) holds in these cases; therefore (17) also must hold. It is really true since $\mathcal{F}(j_0), \dots, \mathcal{F}(j_{n_2-4})$ are $n_2 - 3$ largest levels in X_1. We do not know the monotonicity among the last four u's. An important consequence is that $f^*(j_v)$ cannot be $\lfloor \frac{n_1-n_2}{2} \rfloor$ or $\lceil \frac{n_1+n_2}{2} \rceil$ when $n_2 - 3 > v$.

The above ideas are valid for all cyclic permutations of X_2; therefore $\mathcal{F}_1(B)$ is defined for all $B \subset X_2$ and it is a full level $\binom{X_1}{j}$ for some $j = j(B)(\lfloor \frac{n_1-n_2}{2} \rfloor \le j \le \lceil \frac{n_1+n_2}{2} \rceil)$.

We have to show that $\mathcal{F}_1(B)$ depends only on the size of B, that is, $|B_1| = |B_2|$ implies $\mathcal{F}_1(B_1) = \mathcal{F}_1(B_2)$. It is sufficient to verify this statement for "neighboring" sets, that is, when $|B_1 - B_2| = 1$. Let $B_1 = \{x_1, x_2, \ldots, x_l\}, B_2 = \{x_2, x_3, \ldots, x_l, x_{l+1}\}$. Consider the cyclic permutations $C = (x_2, x_3, \ldots, x_l, x_1, x_{l+1}, x_{l+2} \ldots, x_{n_2}), C' = (x_2, x_3, \ldots, x_l, x_{l+1}, x_1, x_{l+2} \ldots, x_{n_2})$. They define the chains (of length $n_2 + 1$) $\mathcal{L}_1$ and $\mathcal{L}'_1$, which differ only in one member. The function $\mathcal{F}_1$ associates a family $\binom{X_1}{j}(\lfloor\frac{n_1-n_2}{2}\rfloor \le j \le \lceil\frac{n_1+n_2}{2}\rceil)$ with each member of these chains, where the j's are different for one chain. If n_1 and n_2 have the same parities, then there are $n_2 + 1$ choices for j and therefore $\mathcal{F}_1(B_1) = \mathcal{F}_1(B_2)$. If their parities are different, then $\mathcal{F}_1(B_1)$ and $\mathcal{F}_1(B_2)$ may be different: one is $\binom{X_1}{\lfloor\frac{n_1-n_2}{2}\rfloor}$ and the other is $\binom{X_1}{\lceil\frac{n_1+n_2}{2}\rceil}$. It is clear from the monotonicity (17) that this can happen only when $|B_1| = 1$ or $n_2 - 1$. This proves the statement $\mathcal{F}_1(B_1) = \mathcal{F}_1(B_2)$ for $1 < |B_1| = |B_2| < n - 1$. Moreover,

$$\mathcal{F}_1(B) = \text{ either } \binom{X_1}{\lfloor\frac{n_1-n_2}{2}\rfloor} \text{ or } \binom{X_1}{\lceil\frac{n_1+n_2}{2}\rceil} \text{ if } |B| = 1, n-1.$$

Since $\mathcal{F}$ is a two-part Sperner family, $B \subset C$ implies $\mathcal{F}_1(B) \ne \mathcal{F}_1(C)$ (in fact, they must be disjoint). Suppose, e.g., that $j(\{x\}) = \lfloor\frac{n_1-n_2}{2}\rfloor$ holds for some $x \in X_2$. Then $j(C)$ must be $\lceil\frac{n_1+n_2}{2}\rceil$ for all $n_2 - 1$-element C with the possible exception of $X_2 - x$. But these sets cover X_2; therefore $j(\{x\}) = \lfloor\frac{n_1-n_2}{2}\rfloor$ must hold for all $x \in X_2$, and consequently $j(C) = \lceil\frac{n_1+n_2}{2}\rceil$ for all $n_2 - 1$-element $C \in X_2$. We have proved that $\mathcal{F}$ is homogeneous and full, and the function f is defined by $f(i) = j(B)$, where $i = |B|$.

It is almost proved that $\mathcal{F}$ is well-paired, by (17). The only possible exception is that the right-hand side of (2) does not hold for one or more of the pairs $(0, 1), (0, n_2 - 1), (n_2, 1), (n_2, n_2 - 1)$. Suppose, e.g., that the pair $(0, 1)$ is such a one. Then

$$|\mathcal{F}| = \sum_{i=0}^{n_2} \binom{n_2}{i}\binom{n_1}{f(i)}$$

can be increased by interchanging the values $f(0)$ and $f(1)$. (It increases the sum only when $n_2 > 1$ but the case $n_2 = 1$ is trivial.) This contradiction shows that $\mathcal{F}$ is well-paired. □

The interested reader should check [5], where the optimal constructions for all four cases (depending on the parities of n_1 and n_2, resp.,) are illustrated with figures.

3.1. Extremal two-part Sperner families and intervals. Here we give another proof for Theorem 1.1. We need two simple lemmas. Suppose that $u \ge v \ge 1$ are integers, $a_1 \ge a_2 \ge \cdots a_u \ge 0$, $b_1 \ge b_2 \ge \cdots \ge b_v$ are reals, and $g : [v] \to [u]$ is an arbitrary injection (i.e., $g(i) \ne g(j)$ for $i \ne j$). Then we say that the two sequences are *well-paired* by g if $b_i < b_j$ implies $a_{g(i)} \le a_{g(j)}$. Observe that if this definition is applied for the binomial coefficients of ranks n_1 and n_2, respectively, and for the function defined by a homogenous two-part Sperner family, then definition (2) is obtained, again.

LEMMA 3.1. *Suppose that $u \ge v \ge 1$ are integers, $a_1 \ge a_2 \ge \cdots \ge a_u \ge 0$, $b_1 \ge b_2 \ge \cdots \ge b_v$ are reals, and $g : [v] \to [u]$ is an arbitrary injection. Then*

$$\sum_i a_{g(i)} b_i \le \sum_{1 \le i \le v} a_i b_i,$$

and here equality holds if and only if the sequences are well-paired by g.

Lemma 3.2. *Let the $a_1, a_2, \ldots, a_{n_1+1}$ be the sequence of binomial coefficients of rank n_1 in decreasing order, and let $b_1, \ldots, b_{n_2+1}$ be the binomial coefficients of rank n_2 again in decreasing order. (We have $a_i = \binom{n_1}{\lfloor (n_1+i)/2 \rfloor}$ and $b_j = \binom{n_2}{\lfloor (n_2+j)/2 \rfloor}$.) Then $\sum_i a_i b_i = \binom{n}{\lfloor n/2 \rfloor}$.*

Second proof of Theorem 1.1. Let $\mathcal{F}$ be a two-part Sperner family on the n-element underlying set $X = [n]$, with parts X_1, X_2, $|X_i| = n_i$, $n_1 \geq n_2 > 0$. Suppose that $|\mathcal{F}|$ is maximal; then we have $|\mathcal{F}| \geq \binom{n}{\lfloor n/2 \rfloor}$. Let $\pi_i \in S_{[n_i]}$ be a permutation of X_i, $i = 1, 2$. Define the $(n_1+1) \times (n_2+1)$ matrix $M = M(\pi_1, \pi_2)$ as follows. Label the rows by $0, 1, \ldots, n_1$ and the columns by $0, 1, \ldots, n_2$, and for the i, j entry, $M_{i,j}$ equals 1 if the unions of the two initial segments $\{\pi_1(1), \pi_1(2), \ldots, \pi_1(i)\} \cup \{\pi_2(1), \ldots, \pi_2(j)\}$ belong to $\mathcal{F}$, and $M_{i,j} = 0$ for the other entries. Such an M contains at most one nonzero entry in each row and column.

Suppose that M is an arbitrary $(n_1 + 1) \times (n_2 + 1)$ matrix, labeled as above, and suppose that each entry is 0 or 1 and each row and column contains at most one 1. Define a two-part Sperner family $\mathcal{H}(M)$ by taking all sets $F \subset X$ with $M_{|F \cap X_1|, |F \cap X_2|} = 1$. Then $|\mathcal{H}(M)| = \sum_{M_{i,j}=1} \binom{n_1}{i}\binom{n_2}{j}$. By Lemmas 3.1 and 3.2 we have

$$|\mathcal{H}(M)| \leq \sum_{i,j} a_i b_j = \binom{n}{\lfloor n/2 \rfloor}$$

with equality only when M contains a 1 in each column and the mapping defined by M is well-paired with respect the binomial coefficients of ranks n_1 and n_2, respectively.

We obtain

$$\begin{aligned}
|\mathcal{F}| n_1! n_2! \geq \binom{n}{\lfloor n/2 \rfloor} n_1! n_2! &\geq \sum_{(\pi_1, \pi_2)} |\mathcal{H}(M(\pi_1, \pi_2))| \\
&= \sum_{F \in \mathcal{F}} \sum_{\substack{\pi_1, \pi_2 \\ F \cap X_i \text{ is initial in } \pi_i}} \binom{n_1}{|F \cap X_1|}\binom{n_2}{|F \cap X_2|} \\
= \sum_{F \in \mathcal{F}} |F \cap X_1|!(n_1 - |F \cap X_1|)!|F \cap X_2|!(n_2 - |F \cap X_2|)! &\binom{n_1}{|F \cap X_1|}\binom{n_2}{|F \cap X_2|} \\
&= |\mathcal{F}| n_1! n_2!.
\end{aligned}$$

Thus equality holds here, i.e., $|\mathcal{F}| = \binom{n}{\lfloor n/2 \rfloor}$, and so it does for each $|\mathcal{H}(M(\pi_1, \pi_2))|$. It also follows that for each (π_1, π_2), the matrix $M(\pi_1, \pi_2)$ has a 1 in each column and the mapping defined by $M(\pi_1, \pi_2)$ is well-paired. This can be heuristically expressed by saying that the restrictions of $\mathcal{F}$ for a fixed pair of permutations (of X_1 and X_2) is full and well-paired. We have to show that $\mathcal{F}$ is homogeneous, too. In other words, we know that the matrices $M(\pi_1, \pi_2)$ are very similar (there is a little freedom in choosing a 1 in each column), but we have to show that they are identical. Since every permutation can be obtained by interchanging neighboring elements, it is sufficient to show that $M(\pi_1', \pi_2)$ and $M(\pi_1, \pi_2')$ are the same as $M(\pi_1, \pi_2)$ if π_i' is obtained from π_i by interchanging two neighboring elements.

First check what happens if π_2' is obtained from π_2 by interchanging the elements v and $v + 1$ in X_2 ($1 \leq v < n_2$). The initial segments in X_2 are the same for the two permutations π_2 and π_2', except possibly the v-element initial segments. Therefore the new matrices $M = M(\pi_1, \pi_2)$ and $M' = M(\pi_1, \pi_2')$ have the same columns, except eventually the vth one. Since M and M' are full, there are indices u and u' such that $M_{u,v} = 1$ and $M'_{u',v} = 1$. We claim that $u = u'$; the two matrices are identical. Indeed, calculating the cardinalities $|\mathcal{H}(M(\pi_1, \pi_2))|$ and $|\mathcal{H}(M(\pi_1, \pi'_2))|$, both have maximal values. They differ only in the one term, the one containing the factor $\binom{n_2}{v}$. This is multiplied with $\binom{n_1}{u}$ and $\binom{n_1}{u'}$, respectively. Therefore $\binom{n_1}{u} = \binom{n_1}{u'}$ must hold. Hence either $u = u'$ (and we are done) or $u + u' = n_1$. In the latter case consider again the sums

$$\sum_{M_{i,j}=1} \binom{n_1}{i}\binom{n_2}{j} = \sum_{M'_{i,j}=1} \binom{n_1}{i}\binom{n_2}{j}.$$

In the second sum there is no $\binom{n_1}{u}$, and in the first there is no $\binom{n_1}{n_1-u}$. By symmetry, $u < n_1 - u$ can be supposed. By the lemmas, the first sum contains the largest $n_2 + 1$ values of binomial coefficients of rank n_1; this implies that none of $\binom{n_1}{i}(i < u, n_1 - u \leq i$ may occur. On the other hand, all other ones are there: $u \leq i < n_1 - u$. Since the matrix is full, it contains a 1 in each column, and we have $n_1 - 2u = n_2 + 1$ binomial coefficients of rank n_1. The smallest one of them is $\binom{n_1}{u}$. M is well-paired; therefore it must be paired (multiplied) with (one of the) smallest binomial coefficient of rank n_2, namely, $\binom{n_2}{0}$ or $\binom{n_2}{n_2}$. Hence we have $v = 0$ or n_2 in contradiction with the assumption $1 \leq v < n_2$.

Compare now the pairs of permutations (π_1, π_2) and (π'_1, π_2), where π'_1 is obtained from π_1 by interchanging the elements u and $u + 1$ in $X_1(1 \leq u < n_1)$. The matrices $M(\pi_1, \pi_2)$ and $M(\pi'_1, \pi_2)$ are equal except possibly in the uth row. Suppose that both of them have an entry 1 in the uth row and in the vth and in the v'th columns, respectively, where $v \neq v'$. The matrix $M(\pi_1, \pi_2)$ has exactly one 1 in each column, and there is an entry $M_{i,j} = 1$ with $i \neq u, j = v'$. Then $M(\pi'_1, \pi_2)$ has two entries 1 in the v'th column. This contradiction shows $v = v'$; that is, the two matrices are identical. If neither of the two matrices has a 1 in the uth row, then they are the same, again. Finally, if one has a 1 in the uth row and the other one has none, then the sums $\mathcal{H}(M(\pi_1, \pi_2))$ and $\mathcal{H}(M(\pi'_1, \pi_2))$ differ in one positive term; they cannot be (maximally) equal. This contradiction completes the proof of the fact that one change in either permutation does not change the matrix $M(\pi_1, \pi_2)$; they are all the same, and $\mathcal{F}$ is a homogeneous family. □

The interested reader can find further applications of the permutation method in the excellent monograph [2].

REFERENCES

[1] B. Bollobás, *On generalized graphs*, Acta Math. Acad. Sci. Hungar, 16 (1965), pp. 447–452.

[2] K. Engel, *Sperner Theory,* Encyclopedia Math. Appl. 65, Cambridge University Press, Cambridge, UK, 1997.

[3] P. Erdős, *On a lemma of Littlewood and Offord*, Bull. Amer. Math. Soc., 51 (1945), pp. 898–902.

[4] P. L. Erdős, P. Frankl, and G. O. H. Katona, *Extremal hypergraph problems and covex hulls*, Combinatorica, 5 (1985), pp. 11–26.

[5] P. L. Erdős and G. O. H. Katona, *All maximum* 2*-part Sperner families*, J. Combin. Theory Ser. A, 43 (1986), pp. 58–69.

[6] Z. Füredi, J. R. Griggs, A. M. Odlyzko, and J. M. Shearer, *Ramsey–Sperner theory*, Discrete Math., 63 (1987), pp. 143–152.

[7] G. O. H. Katona, *On a conjecture of Erdős and a stronger form of Sperner's theorem*, Stud. Sci. Math. Hungar., 1 (1966), pp. 59–63.

[8] G. O. H. Katona, *A simple proof of the Erdős-Chao Ko-Rado theorem*, J. Combin. Theory Ser. B, 13 (1972), pp. 183–184.

[9] G. O. H. Katona, *Extremal problems for hypergraphs*, in Combinatorics, Proceedings of the NATO Advanced Study Institute, Breukelen, 1974, Part 2, Math. Centre Tracts 56, Math. Centrum, Amsterdam, 1974, pp. 13–42.

[10] G. O. H. Katona, *The cycle method and its limits*, in Numbers, Information, and Complexity, I. Althöfer, Ning Cai, G. Dueck, L. Khachatrian, M. S. Pinsker, A. Sárközy, I. Wegener, and Zh. Zhang, eds., Kluwer, Dordrecht, The Netherlands, 2000, pp. 129–141.

[11] D. J. Kleitman, *On a lemma of Littlewood and Offord on the distribution of certain sums*, Math. Z., 90 (1965), pp. 251–259.

[12] D. Lubell, *A short proof of Sperner's lemma*, J. Combin. Theory, 1 (1966), p. 299.

[13] L. D. Meshalkin, *A generalization of Sperner's theorem on the number of a finite set*, in Russian, Teor. Verojatnost. Primen., 8 (1963), pp. 219–220.

[14] S. Shahriari, *On the structure of maximum 2-part Sperner families*, Discrete Math., 162 (1996), pp. 229–238.

[15] E. Sperner, *Ein Satz über Untermegen einer endlichen Menge*, Math. Z., 27, (1928), pp. 544–548.

[16] K. Yamamoto, *Logarithmic order of free distributive lattices*, J. Math. Soc. Japan, 6 (1954), pp. 347–357.

SIAM J. DISCRETE MATH.
Vol. 19, No. 2, pp. 501–522

POLYHEDRAL ANALYSIS FOR THE UNCAPACITATED HUB LOCATION PROBLEM WITH MODULAR ARC CAPACITIES[*]

HANDE YAMAN[†]

Abstract. We consider the problem of installing a two-level telecommunication network. Terminal nodes communicate with each other through hubs. Hubs can be installed on terminal nodes and they are interconnected by a complete network. Each terminal is connected directly to a hub node. Integer amounts of capacity units are installed on the arcs between hub pairs and terminals and their hubs. The aim is to minimize the cost of installing hubs and capacity units on arcs. We present valid and facet defining inequalities for the polyhedron associated with this problem.

Key words. hub location, polyhedral analysis, lifting

AMS subject classifications. 90C10, 90C57, 90B80

DOI. 10.1137/S0895480103439157

1. Introduction.

We consider the problem of locating hubs in a telecommunication network. Hubs (servers, concentrators, etc.) are installed to route the traffic of terminals (users). Given a set of terminals, a subset is chosen to be the set of hub locations. Each terminal that does not become a hub is directly connected to a single hub. The network connecting the hubs is called the *backbone network* and a network connecting the terminals to a hub is called a *local access network (LAN)*. We consider telecommunication networks where the backbone is complete and the LANs are stars.

The traffic between two terminals goes from the origin terminal to its hub, then to the hub of the destination terminal, and then to the destination itself. So the total traffic on the arc from a terminal to its hub is the traffic originating at that terminal node, and the traffic on the arc from a hub to a terminal connected to that hub is the traffic arriving at that terminal node. The total traffic to travel from hub j to hub l is the traffic from terminals connected to hub j to terminals connected to hub l. The traffic flows on arcs and capacity units can be installed on arcs in integer amounts.

In Figure 1.1, we see a network with three hubs. The traffic between any two nodes is 0.5 and the capacity unit is 1 on all arcs. The amount of capacity units to be installed on the arcs are given in the figure. For example, we need to install $\lceil 7 \times 0.5 \rceil = 4$ capacity units on an arc from a terminal to its hub.

The cost of installing such a telecommunication network is the sum of the cost of locating hubs and the cost of installing capacity units on arcs. The *uncapacitated hub location problem with modular arc capacities (HLM)* is the problem of locating hubs and connecting the remaining nodes to hubs with the aim of minimizing this total cost. Labbé and Yaman [11] prove that the special case of HLM where the cost of installing capacity units on the backbone network is zero is NP-hard.

Campbell, Ernst, and Krishnamoorthy [3] give a survey of hub location problems. Klincewicz [7] gives a survey of hub location problems in telecommunications.

Very little is known about the polyhedra associated with hub location problems. A similar problem with no cost for installing capacity units on arcs but a cost for routing

[*] Received by the editors December 30, 2003; accepted for publication (in revised form) January 3, 2005; published electronically November 4, 2005.
http://www.siam.org/journals/sidma/19-2/43915.html

[†] Bilkent University, Department of Industrial Engineering, Bilkent, 06800 Ankara, Turkey (hyaman@bilkent.edu.tr).

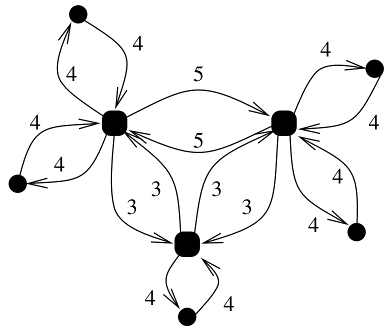


FIG. 1.1. *A network with three hubs.*

the traffic is called the *uncapacitated hub location problem with single assignment (HLs).* Polyhedral analysis for this problem can be found in [12] and [10]. If, in addition, we allow a terminal to be connected to several hubs, then the problem is called the *uncapacitated hub location problem with multiple assignment (HLm).* Polyhedral properties of HLm are studied by Hamacher et al. [6].

Chung, Myung, and Tcha [5] study a version of HLM where there is a fixed cost of establishing a link between two hubs. In HLM, this corresponds to the case where backbone links are uncapacitated, meaning that if two nodes become hubs, only one capacity unit is installed between them. The authors propose a branch and cut algorithm for this problem.

Yaman and Carello [15] consider a generalization of HLM where hubs are capacitated; the amount of traffic transiting through a hub is limited by the capacity of the hub. They present a metaheuristic and a branch and cut algorithm to solve this problem. Their branch and cut algorithm uses cuts given in [12] and [10].

In this paper, we present valid and facet defining inequalities for the polyhedron associated with HLM. We give several lifting results which can be used to derive further facet defining inequalities. The paper is organized as follows. In section 2, we give a formulation of the problem. We present valid inequalities in section 3. Section 4 is devoted to polyhedral analysis. We conclude in section 5.

2. Formulation.

Let I denote the set of terminal nodes with $|I| = n$. Any distinct pair of terminal nodes defines a commodity. We denote by K the set of commodities. For commodity $(i, m) \in K$, i is the origin, m is the destination, and t_{im} is the amount of traffic to be routed from i to m. We define t_{ii} to be 0 for all $i \in I$.

Each terminal either becomes a hub or is connected to another node which becomes a hub. The cost of installing a hub at node $i \in I$ is denoted by C_{ii}. Hubs are connected by a complete directed graph. Each nonhub node is directly connected to its hub. Integer amounts of capacity are installed on the arcs between pairs of hubs and between terminals and their hubs. We assume that the capacity unit on all arcs is 1 and that the demands are scaled accordingly. The capacity of each terminal-hub and hub-terminal arc is fully determined by the chosen terminal-hub connection. The cost of connecting node $i \in I$ to node $j \in I \setminus \{i\}$, denoted by C_{ij}, is equal to the cost of installing $\lceil \sum_{m \in I} t_{im} \rceil + \lceil \sum_{m \in I} t_{mi} \rceil$ capacity units between nodes i and j.

We define the arc set $A = \{(j, l) : j \in I, l \in I, j \neq l\}$. We denote by R_{jl} the cost of installing a capacity unit on arc (j, l) if it becomes a backbone arc. Let K'_{jl} be the set of commodities (i, m) such that i is connected to j and m is connected to

l. If nodes j and l become hubs, then the amount of flow on arc (j,l) is given by $\sum_{(i,m)\in K'_{jl}} t_{im}$ and $\lceil \sum_{(i,m)\in K'_{jl}} t_{im} \rceil$ units of capacity should be installed on this arc.

We define the assignment variable x_{ij} to be 1 if terminal $i \in I$ is assigned (connected) to hub $j \in I$ and 0 otherwise. If node i becomes a hub, then x_{ii} is 1. We further define z_{jl} to be the amount of capacity units installed on arc $(j,l) \in A$.

The HLM can be formulated as follows (see [12]):

$$\min \sum_{i\in I}\sum_{j\in I} C_{ij}x_{ij} + \sum_{(j,l)\in A} R_{jl}z_{jl} \tag{2.1}$$

$$\text{subject to} \sum_{j\in I} x_{ij} = 1 \qquad \forall i \in I, \tag{2.2}$$

$$x_{ij} \le x_{jj} \qquad \forall (i,j) \in A, \tag{2.3}$$

$$z_{jl} \ge \sum_{(i,m)\in K'} t_{im}(x_{ij} + x_{ml} - 1) \qquad \forall (j,l) \in A, K' \subseteq K, \tag{2.4}$$

$$z_{jl} \text{ integer} \qquad \forall (j,l) \in A, \tag{2.5}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in I, j \in I. \tag{2.6}$$

Constraints (2.2), (2.3), and (2.6) ensure that each terminal either becomes a hub or is assigned to exactly one hub. Constraints (2.4) relate the capacity vector z to the assignment vector x. For arc $(j,l) \in A$, because of constraints (2.5) and (2.6), constraint set (2.4) is equivalent to

$$z_{jl} \ge \left\lceil \max_{K'\subseteq K} \left(\sum_{(i,m)\in K'} t_{im}(x_{ij} + x_{ml} - 1) \right) \right\rceil = \left\lceil \sum_{(i,m)\in K'_{jl}} t_{im} \right\rceil.$$

If $R_{jl} > 0$, then an optimal solution satisfies the inequality at equality.

The objective function (2.1) consists of the cost of locating hubs and the cost of installing capacity units on arcs.

3. Valid inequalities. In this section, we present families of valid inequalities for the polyhedron associated with HLM and point out the domination relations among these valid inequalities. We investigate inequalities that involve both the assignment and the capacity variables.

DEFINITION 3.1. *Let*

$$F = \left\{ (x,z) \in \{0,1\}^{n^2} \times \mathbb{Z}^{n(n-1)} : (x,z) \textit{ satisfies } (2.2)\text{–}(2.6) \right\}$$

and

$$P = \text{conv}(F).$$

Labbé, Yaman, and Gourdin [12] study the HLs which is obtained by relaxing integrality constraints (2.5) in HLM. They derive valid inequalities by projecting out the flow variables in a larger formulation for this relaxed problem. These inequalities are given in the following proposition.

PROPOSITION 3.2 (Labbé, Yaman, and Gourdin [12]). *Let S and T be nonempty disjoint subsets of I and $K' \subseteq K$. The projection inequality*

$$\sum_{j\in S}\sum_{l\in T} z_{jl} \ge \sum_{(i,m)\in K'} t_{im} \left(\sum_{j\in S} x_{ij} + \sum_{l\in T} x_{ml} - 1 \right) \tag{3.1}$$

is valid for P.

Constraints (2.4) are projection inequalities where sets S and T are singletons.

Projection inequalities (3.1) ignore the integrality of z_{jl} variables. Now we present a family of inequalities which use this information.

For $K^{'} \subseteq K$, let

$$O(K^{'}) = \{i \in I : \exists m \in I \setminus \{i\} \text{ with } (i, m) \in K^{'}\}$$

and

$$D(K^{'}) = \{i \in I : \exists m \in I \setminus \{i\} \text{ with } (m, i) \in K^{'}\}.$$

Proposition 3.3. *Let* S *and* T *be nonempty disjoint subsets of* I *and* $K^{'} \subseteq K$. *Inequality*

$$\sum_{j\in S}\sum_{l\in T} z_{jl} \geq \left\lceil \sum_{(i,m)\in K'} t_{im} \right\rceil \left(1 - \sum_{i\in O(K')}\sum_{j\in I\setminus S} x_{ij} - \sum_{m\in D(K')}\sum_{l\in I\setminus T} x_{ml}\right) \tag{3.2}$$

is valid for P.

Proof. For $(x, z) \in F$, the right-hand side of inequality (3.2) is $\lceil\sum_{(i,m)\in K'} t_{im}\rceil$ if $\sum_{j\in I\setminus S} x_{ij} = 0$ for all $i \in O(K^{'})$ and $\sum_{l\in I\setminus T} x_{ml} = 0$ for all $m \in D(K^{'})$. It is nonpositive otherwise. □

Notice that different sets $K^{'}$ can lead to the same sets $O(K^{'})$ and $D(K^{'})$. For a given fractional solution, it is important to be able to choose among these subsets $K^{'}$ the one which leads to the most violated inequality.

For subsets O and D of I, let

$$\kappa(O, D) = \big\{(i, m) \in K : i \in O \text{ and } m \in D\big\}.$$

Proposition 3.4. *Let* (x, z) *be a fractional solution which satisfies constraints* (2.2). *If there exists an inequality* (3.2) *violated by* (x, z), *then there exists a violated inequality* (3.2) *for some* $K^{'} \subseteq K$ *such that* $O(K^{'}) \cap D(K') = \emptyset$ *and* $K^{'} = \kappa(O(K^{'}), D(K^{'}))$.

Proof. For $K^{'} \subseteq K$, if $|O(K^{'}) \cap D(K^{'})| \geq 1$, then

$$\begin{aligned}
&1 - \sum_{i\in O(K')}\sum_{j\in I\setminus S} x_{ij} - \sum_{m\in D(K')}\sum_{l\in I\setminus T} x_{ml} \\
&= \sum_{i\in O(K')}\sum_{j\in S} x_{ij} + \sum_{m\in D(K')}\sum_{l\in T} x_{ml} - |O(K^{'})| - |D(K^{'})| + 1 \\
&= \sum_{i\in O(K')\setminus D(K')}\sum_{j\in S} x_{ij} - |O(K^{'}) \setminus D(K^{'})| + \sum_{m\in D(K')\setminus O(K')}\sum_{l\in T} x_{ml} - |D(K^{'}) \setminus O(K^{'})| \\
&\qquad + \sum_{i\in O(K')\cap D(K')}\sum_{j\in S\cup T} x_{ij} - 2|O(K^{'}) \cap D(K^{'})| + 1 \\
&\leq \sum_{i\in O(K')\cap D(K')}\sum_{j\in S\cup T} x_{ij} - 2|O(K^{'}) \cap D(K^{'})| + 1 \\
&\leq (-|O(K^{'}) \cap D(K^{'})| + 1) \leq 0.
\end{aligned}$$

Therefore, inequality (3.2) for this choice of $K^{'}$ cannot be violated. This proves that if inequality (3.2) is violated for $K^{'}$, then $O(K^{'}) \cap D(K^{'}) = \emptyset$. The second part of the proposition is then trivial. □

If S and T are singletons, then inequality (3.2) becomes

$$z_{jl} \geq \left\lceil \sum_{(i,m)\in K'} t_{im} \right\rceil \left(1 - \sum_{i\in O(K')} \sum_{u\in I\setminus\{j\}} x_{iu} - \sum_{m\in D(K')} \sum_{u\in I\setminus\{l\}} x_{mu} \right). \tag{3.3}$$

If, in the formulation (2.1)–(2.6), we replace constraints (2.4) and (2.5) with the set of inequalities (3.3) for all disjoint subsets O and D of I, $K^{'} = \kappa(O, D)$, and $(j, l) \in A$, we obtain a valid formulation for HLM where we do not need to impose explicitly the integrality of z_{jl} variables. For $(j, l) \in A$, constraints (2.4) linearize the nonlinear requirement

$$z_{jl} \geq \sum_{(i,m)\in K} t_{im} x_{ij} x_{ml}$$

by linearizing the equivalent family of nonlinear inequalities

$$z_{jl} \geq \sum_{(i,m)\in K'} t_{im} x_{ij} x_{ml}$$

for all $K^{'} \subseteq K$. Inequalities (3.3) linearize the nonlinear requirement

$$z_{jl} \geq \left\lceil \sum_{(i,m)\in K} t_{im} x_{ij} x_{ml} \right\rceil$$

by linearizing the equivalent family of nonlinear inequalities

$$z_{jl} \geq \left\lceil \sum_{(i,m)\in \kappa(O,D)} t_{im} \right\rceil \Pi_{i\in O} x_{ij} \Pi_{m\in D} x_{ml}$$

for all disjoint subsets O and D of I.

The following example shows that it is not possible to compare the LP relaxations of these two formulations.

Example 3.1. Comparing the LP relaxation of formulation (2.1)–(2.6) with that of formulation (2.1)–(2.3), (2.6), and (3.3) is equivalent to comparing the relative strength of inequalities (2.4) and (3.3). Let $I = \{1, 2, 3, 4\}$. Consider a vector x such that $x_{12} = x_{22} = x_{34} = x_{44} = 0.6$ and $x_{11} = x_{21} = x_{33} = x_{43} = 0.4$ (see Figure 3.1).

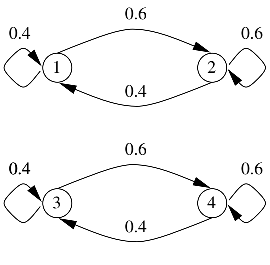


FIG. 3.1. *Example* 3.1*: assignment of nodes.*

For arc (2, 4) and $K^{'} = \{(1, 3), (1, 4), (2, 3), (2, 4)\}$, constraint (2.4) reads

$$z_{24} \geq 0.2(t_{13} + t_{14} + t_{23} + t_{24}). \tag{3.4}$$

This is indeed a best choice of $K^{'}$ in the sense that it can lead to a most violated inequality (2.4) for arc $(2,4)$.

Now we consider inequality (3.3) for arc $(2,4)$. Let O and D be disjoint subsets of I. To find a most violated inequality, it is better to choose $O \subseteq \{1,2\}$ and $D \subseteq \{3,4\}$. Then, inequality (3.3) is $z_{24} \geq \lceil \sum_{(i,m)\in\kappa(O,D)} t_{im} \rceil (1 - 0.4|O| - 0.4|D|)$. The right-hand side of this inequality can be positive only if $|O| = |D| = 1$. If $|O| = |D| = 1$ and $i \in O$ and $m \in D$, then the inequality is

$$z_{24} \geq 0.2\lceil t_{im} \rceil. \tag{3.5}$$

The best inequality can be obtained by choosing a commodity (i,m) with maximum t_{im}. Assume without loss of generality that this maximum is attained at $i = 1$ and $m = 3$.

If $t_{13} + t_{14} + t_{23} + t_{24} > \lceil t_{13} \rceil$, then inequality (3.4) imposes a higher lower bound than inequality (3.5). And if $t_{13}+t_{14}+t_{23}+t_{24} < \lceil t_{13} \rceil$, then the lower bound imposed by inequality (3.5) is higher than the one imposed by inequality (3.4). Therefore, these two inequalities are not comparable.

For given sets S and T, inequalities (3.1) can be separated in polynomial time (see [12]). However, the complexity of the separation of inequalities (3.2) is open even when S and T are given. Still, the separation is easy if x is not fractional. In this case, sets S and T should be singletons and

$$K^{'} = \left\{ (i,m) \in K, \sum_{j\in S} x_{ij} = 1 \text{ and } \sum_{l\in T} x_{ml} = 1 \right\}.$$

Yaman and Carello [15] present inequalities that dominate the projection inequalities (3.1).

Proposition 3.5 (Yaman and Carello [15]). *Let S and T be nonempty disjoint subsets of I and $K^{'} \subseteq K$. The improved projection inequality*

$$\begin{aligned}
\sum_{j\in S}\sum_{l\in T} z_{jl} \geq & \sum_{(i,m)\in K':i\notin S, m\notin T} t_{im}\left(\sum_{j\in S\setminus\{m\}} x_{ij} + \sum_{l\in T\setminus\{i\}} x_{ml} + x_{im} + x_{mi} - 1 \right) \\
& + \sum_{(i,m)\in K':i\in S, m\notin T} t_{im}\left(\sum_{j\in S\setminus\{m\}} x_{ij} + \sum_{l\in T} x_{ml} + x_{im} - 1 \right) \\
& + \sum_{(i,m)\in K':i\notin S, m\in T} t_{im}\left(\sum_{j\in S} x_{ij} + \sum_{l\in T\setminus\{i\}} x_{ml} + x_{mi} - 1 \right) \\
& + \sum_{(i,m)\in K':i\in S, m\in T} t_{im}\left(\sum_{j\in S} x_{ij} + \sum_{l\in T} x_{ml} - 1 \right)
\end{aligned}$$

is valid for P.

We present inequalities that dominate inequalities (3.2) in the same manner.

Proposition 3.6. *Let S, T, O, and D be nonempty subsets of I such that*

$S \cap T = \emptyset$ and $O \cap D = \emptyset$. Inequality

$$\begin{aligned}\sum_{j\in S}\sum_{l\in T} z_{jl} \geq \left\lceil \sum_{(i,m)\in\kappa(O,D)} t_{im} \right\rceil \Bigg[\sum_{i\in O}\left(\sum_{j\in S} x_{ij} + \sum_{m\in D\setminus(S\cup T)} x_{im} - 1 \right) \\ (3.6) \qquad + \sum_{m\in D}\left(\sum_{l\in T} x_{ml} + \sum_{i\in O\setminus(S\cup T)} x_{mi} - 1 \right) + 1 \Bigg]\end{aligned}$$

is valid for P.

Proof. If $\sum_{m\in D\setminus(S\cup T)} x_{im} = 0$ for all $i \in O$ and $\sum_{i\in O\setminus(S\cup T)} x_{mi} = 0$ for all $m \in D$, then inequality (3.6) reduces to inequality (3.2) for $K^{'} = \kappa(O, D)$.

If there exists $i \in O$ and $m \in D \setminus (S \cup T)$ such that $x_{im} = 1$ (resp., $m \in D$ and $i \in O \setminus (S \cup T)$ such that $x_{mi} = 1$), then as $x_{mm} = 1$, $m \notin T$ and $m \notin O$, we have $\sum_{l\in T} x_{ml} + \sum_{l\in O\setminus(S\cup T)} x_{ml} = 0$ (resp., $\sum_{j\in S} x_{ij} + \sum_{j\in D\setminus(S\cup T)} x_{ij} = 0$). This implies that the right-hand side of inequality (3.6) is nonpositive. □

Inequality (3.6) remains valid if $\lceil\sum_{(i,m)\in\kappa(O,D)} t_{im}\rceil$ is changed to $\lceil\sum_{(i,m)\in K'} t_{im}\rceil$ for $K^{'} \subset \kappa(O, D)$. But these new inequalities are dominated.

PROPOSITION 3.7. *For given nonempty subsets S, T, O, and D of I such that $S \cap T = \emptyset$ and $O \cap D = \emptyset$, inequality* (3.6) *dominates inequality* (3.2).

Proof. If $K^{'} = \kappa(O, D)$, then inequality (3.6) dominates inequality (3.2). If $K^{'} \neq \kappa(O, D)$, then by Proposition 3.4, inequality (3.2) for $\kappa(O, D)$ dominates inequality (3.2) for $K^{'}$. □

In inequality (3.2), when a node in $O(K^{'})$ is assigned to some node in $I \setminus S$ or a node in $D(K^{'})$ is assigned to some node in $I \setminus T$, the right-hand side of the inequality is nonpositive, since the coefficients of the assignment variables are all equal to $\lceil\sum_{(i,m)\in K'} t_{im}\rceil$. In the remaining part of this section, we present families of valid inequalities where the assignment variables have smaller coefficients so that even when there exist nodes in $O(K^{'})$ which are assigned to nodes in $I \setminus S$ or nodes in $D(K^{'})$ which are assigned to nodes in $I \setminus T$, the inequality can still give a positive lower bound on $\sum_{j\in S}\sum_{l\in T} z_{jl}$.

PROPOSITION 3.8. *Let S and T be nonempty disjoint subsets of I and $K^{'} \subseteq K$. Inequality*

$$\begin{aligned}\sum_{j\in S}\sum_{l\in T} z_{jl} \geq \left\lceil \sum_{(i,m)\in K'} t_{im} \right\rceil - \sum_{i\in O(K')}\left(\left\lceil \sum_{m:(i,m)\in K'} t_{im} \right\rceil \sum_{j\in I\setminus S} x_{ij} \right) \\ (3.7) \qquad - \sum_{m\in D(K')}\left(\left\lceil \sum_{i:(i,m)\in K'} t_{im} \right\rceil \sum_{l\in I\setminus T} x_{ml} \right)\end{aligned}$$

is valid for P.

Proof. For a given x, define $O^{'} = \{i \in O(K^{'}) : \sum_{j\in I\setminus S} x_{ij} = 0\}$ and $D^{'} = \{m \in D(K^{'}) : \sum_{l\in I\setminus T} x_{ml} = 0\}$. Then the right-hand side of inequality (3.7) is equal to

$$\begin{aligned}\left\lceil \sum_{(i,m)\in K'} t_{im} \right\rceil - \sum_{i\in O(K')\setminus O'} \left\lceil \sum_{m:(i,m)\in K'} t_{im} \right\rceil - \sum_{m\in D(K')\setminus D'} \left\lceil \sum_{i:(i,m)\in K'} t_{im} \right\rceil \\ \leq \left\lceil \sum_{(i,m)\in K' : i\in O' \text{ and } m\in D'} t_{im} \right\rceil \leq \left\lceil \sum_{i\in O'}\sum_{m\in D'} t_{im} \right\rceil.\end{aligned}$$

The last term is a valid lower bound on $\sum_{j\in S}\sum_{l\in T} z_{jl}$. □

Different from inequalities (3.2) and (3.6), inequalities (3.7) defined by sets $K^{'} \neq \kappa(O,D)$ can be nondominated. If there exists a commodity $(u,v)\in\kappa(O,D)$ such that $\lceil\sum_{(i,m)\in\kappa(O,D)} t_{im}\rceil = \lceil\sum_{(i,m)\in\kappa(O,D)\setminus\{(u,v)\}} t_{im}\rceil$, then inequality (3.7) for $\kappa(O,D)\setminus\{(u,v)\}$ either is the same as inequality (3.7) for $\kappa(O,D)$ or dominates it. An example is given.

Example 3.2. Let $I=\{1,2,3,4\}$. The nonzero traffic values are as follows: $t_{13}=1.25$, $t_{14}=1$, $t_{23}=1.95$, $t_{24}=0.05$ (see Figure 3.2).

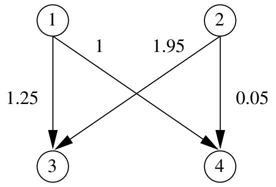


FIG. 3.2. *Example* 3.2*: nonzero traffic values.*

We consider some arc (j,l). Inequality (3.7) for $\kappa(O,D)$, where $O=\{1,2\}$ and $D=\{3,4\}$, is

$$\begin{aligned} z_{jl} \geq\ & \lceil t_{13}+t_{14}+t_{23}+t_{24}\rceil - \lceil t_{13}+t_{14}\rceil(1-x_{1j}) - \lceil t_{23}+t_{24}\rceil(1-x_{2j}) \\ & -\lceil t_{13}+t_{23}\rceil(1-x_{3l}) - \lceil t_{14}+t_{24}\rceil(1-x_{4l}) \\ =\ & 5-3(1-x_{1j})-2(1-x_{2j})-4(1-x_{3l})-2(1-x_{4l}). \end{aligned}$$

For $K^{'}=\{(1,3),(1,4),(2,3)\}$, inequality (3.7) is

$$\begin{aligned} z_{jl} \geq\ & \lceil t_{13}+t_{14}+t_{23}\rceil - \lceil t_{13}+t_{14}\rceil(1-x_{1j}) - \lceil t_{23}\rceil(1-x_{2j}) \\ & -\lceil t_{13}+t_{23}\rceil(1-x_{3l}) - \lceil t_{14}\rceil(1-x_{4l}) \\ =\ & 5-3(1-x_{1j})-2(1-x_{2j})-4(1-x_{3l})-1(1-x_{4l}). \end{aligned}$$

Inequality (3.7) for $K^{'}$ dominates inequality (3.7) for $\kappa(O,D)$.

The complexity of the separation is open for inequalities (3.7). If one approximates the separation problem by removing the ceilings, then the new problem is the same as the separation problem for projection inequalities (3.1).

The coefficients of some variables can be further improved as follows.

PROPOSITION 3.9. *Let S and T be nonempty disjoint subsets of I and $K^{'}\subseteq K$. For $i^*\in O(K^{'})$, inequality*

$$\begin{aligned} \sum_{j\in S}\sum_{l\in T} z_{jl} \geq & \left\lceil \sum_{(i,m)\in K'} t_{im}\right\rceil - \sum_{i\in O(K')\setminus i^*}\left(\left\lceil\sum_{m:(i,m)\in K'} t_{im}\right\rceil \sum_{j\in I\setminus S} x_{ij}\right) \\ & -\left(\left\lceil\sum_{(i,m)\in K'} t_{im}\right\rceil - \left\lceil \sum_{(i,m)\in K'} t_{im} - \sum_{m:(i^*,m)\in K'} t_{i^*m}\right\rceil\right)\sum_{j\in I\setminus S} x_{i^*j} \\ & - \sum_{m\in D(K')}\left(\left\lceil\sum_{i:(i,m)\in K'} t_{im}\right\rceil \sum_{l\in I\setminus T} x_{ml}\right) \end{aligned} \tag{3.8}$$

is a valid inequality for P. Similarly, for $i^* \in D(K^{'})$, inequality

$$\begin{aligned}\sum_{j\in S}\sum_{l\in T} z_{jl} \ge{}& \left\lceil \sum_{(i,m)\in K'} t_{im}\right\rceil - \sum_{i\in O(K')}\left(\left\lceil \sum_{m:(i,m)\in K'} t_{im}\right\rceil \sum_{j\in I\setminus S} x_{ij}\right)\\ &-\left(\left\lceil \sum_{(i,m)\in K'} t_{im}\right\rceil - \left\lceil \sum_{(i,m)\in K'} t_{im} - \sum_{m:(m,i^*)\in K'} t_{mi^*}\right\rceil\right)\sum_{l\in I\setminus T} x_{i^*l}\\ &-\sum_{m\in D(K')\setminus i^*}\left(\left\lceil \sum_{i:(i,m)\in K'} t_{im}\right\rceil \sum_{l\in I\setminus T} x_{ml}\right)\end{aligned} \tag{3.9}$$

is valid for P.

Proof. We prove the validity of inequality (3.8). Validity of inequality (3.9) can be proved in a similar way. If $\sum_{j\in I\setminus S} x_{i^*j} = 0$, then inequality (3.8) is the same as inequality (3.7). If $\sum_{j\in I\setminus S} x_{i^*j} = 1$, then it is dominated by inequality (3.7) for $K^{''} = \{(i,m) \in K^{'} : i \neq i^*\}$. □

To conclude this section, we compare inequalities (3.2), (3.7), (3.8), and (3.9).

PROPOSITION 3.10. *For given nonempty disjoint subsets S and T of I and $K^{'} \subseteq K$, inequality* (3.7) *dominates inequality* (3.2) *and inequalities* (3.8) *and* (3.9) *dominate inequality* (3.7).

4. Facet defining inequalities.

This section is devoted to the polyhedral analysis for the HLM polyhedron. We first prove some properties of the facet defining inequalities and then present families of such inequalities.

4.1. Basics.

We reformulate the problem by substituting $x_{jj} = 1-\sum_{m\in I\setminus\{j\}} x_{jm}$ for all $j \in I$ (see Avella and Sassano [1]). We also eliminate some inequalities (2.4). If both j and l become hubs, then the traffic of commodities with destination j or origin l does not travel on arc (j,l). Moreover, the traffic from node j to node l travels on arc (j,l). Define for $(j,l) \in A$,

$$K_{jl} = K \setminus \left(\{(j,l)\} \cup \{(m,j) : m \in I \setminus \{j\}\} \cup \{(l,m) : m \in I \setminus \{l\}\}\right).$$

The HLM can be reformulated as follows:

$$\begin{aligned}&\min \sum_{i\in I}\sum_{j\in I\setminus\{i\}} C_{ij}x_{ij} + \sum_{i\in I} C_{ii}\left(1-\sum_{j\in I\setminus\{i\}} x_{ij}\right) + \sum_{(j,l)\in A} R_{jl}z_{jl}\\ &\text{s.t. } x_{ij} + \sum_{m\in I\setminus\{j\}} x_{jm} \le 1 \qquad \forall (i,j)\in A,\\ &\quad z_{jl} \ge \sum_{(i,m)\in K',i\neq j,m\neq l} t_{im}(x_{ij}+x_{ml}-1)\\ &\qquad + \sum_{i\in I:(j,i)\in K'} t_{ji}\left(x_{il} - \sum_{m\in I\setminus\{j\}} x_{jm}\right)\\ &\qquad + \sum_{i\in I:(i,l)\in K'} t_{il}\left(x_{ij} - \sum_{m\in I\setminus\{l\}} x_{lm}\right)\end{aligned} \tag{4.1}$$

$$+\, t_{jl}\Bigg(1-\sum_{m\in I\setminus\{j\}} x_{jm}-\sum_{m\in I\setminus\{l\}} x_{lm}\Bigg) \qquad \forall K^{'}\subseteq K_{jl},(j,l)\in A, \tag{4.2}$$

$$x_{ij}\in\{0,1\} \qquad \forall (i,j)\in A, \tag{4.3}$$

$$z_{jl}\geq 0 \qquad \forall (j,l)\in A, \tag{4.4}$$

$$z_{jl}\ \text{integer} \qquad \forall (j,l)\in A. \tag{4.5}$$

Definition 4.1. *Let*

$$P_A=\operatorname{conv}\big(\{(x,z)\in\{0,1\}^{n(n-1)}\times\mathbb{Z}^{n(n-1)}:(x,z)\ \textit{satisfies}\ (4.1)\text{–}(4.5)\}\big).$$

Define also

$$P_\emptyset=\operatorname{conv}\big(\{x\in\{0,1\}^{n(n-1)}:x\ \textit{satisfies}\ (4.1)\ \textit{and}\ (4.3)\}\big).$$

Polytope $P_\emptyset$ is a special stable set polytope. (See, e.g., [2], [4], and [14] for polyhedral properties of the stable set polytope and see [9] for facet defining inequalities of $P_\emptyset$.) Polytope $P_\emptyset$ is interesting since $P_\emptyset=Proj_x(P_A)$. Labbé and Yaman [8] describe the relationship between the facets of $P_\emptyset$ and P_A. The following two propositions are corollaries of the results in [8] and the proofs can be found in that paper. Similar results are also proved by Labbé, Yaman, and Gourdin [12] for the polyhedron associated with HLs.

Proposition 4.2. *The polyhedron P_A is full dimensional, i.e.,* $\dim(P_A)=2n(n-1)$.

Proposition 4.3. *The inequality $\pi x\leq\pi_0$ defines a facet of P_A if and only if it defines a facet of $P_\emptyset$.*

This proposition gives a characterization of the facet defining inequalities of P_A which involve only the assignment variables, in terms of the facet defining inequalities of $P_\emptyset$. Next, we investigate facet defining inequalities of P_A which involve only the capacity variables. The proofs of the following two propositions are similar to the proofs of Proposition 4.3 and 4.4 in [12] and are omitted here.

Proposition 4.4. *Every facet defining inequality of P_A of the form $\beta z\geq\beta_0$ is a positive multiple of $z_{jl}\geq 0$ for some $(j,l)\in A$.*

This proposition implies that it is not possible to find fixed positive lower bounds on capacity variables. This is natural since if all nodes are assigned to the same hub, then there is no traffic in the backbone network.

Proposition 4.5. *For $(j,l)\in A$, if $t_{jl}=0$, then the inequality $z_{jl}\geq 0$ defines a facet of P_A.*

4.2. General lifting results. In what follows, we give some properties of facet defining inequalities that involve both the assignment and the capacity variables.

Define $e^x_{ij}=(x,z)$ (resp., $e^z_{ij}=(x,z)$) to be the unit vector such that $x_{lm}=0$ for all $(l,m)\in A\setminus\{(i,j)\}$, $x_{ij}=1$ and $z_{lm}=0$ for all $(l,m)\in A$ (resp., $x_{lm}=0$ for all $(l,m)\in A$, $z_{lm}=0$ for all $(l,m)\in A\setminus\{(i,j)\}$ and $z_{ij}=1$).

Definition 4.6. *For $B\subseteq A$, define*

$$\begin{aligned}F_B=&\big\{(x,z)\in\{0,1\}^{n(n-1)}\times\mathbb{Z}^{|B|}:(x,z)\ \textit{satisfies}\ (4.1)\ \textit{and}\ (4.3)\ \forall(i,j)\in A\\ &\qquad \textit{and}\ (4.2),\ (4.4),\ \textit{and}\ (4.5)\ \forall(j,l)\in B\big\}\end{aligned}$$

and let

$$P_B=\operatorname{conv}(F_B).$$

If $B = \{(j,l)\}$, then we write F_{jl} and P_{jl} for F_B and P_B, respectively.

In other words, P_B is the projection of P_A on the space of x_{ij} for all $(i,j) \in A$ and z_{jl} for all $(j,l) \in B$. Facet defining inequalities of P_B and P_A are related in the following way.

THEOREM 4.7. *For $B \subset A$, inequality $\beta z \geq \alpha x + \pi$ with $\beta_{jl} = 0$ for all $(j,l) \in A \setminus B$ is facet defining for P_A if and only if it is facet defining for P_B.*

Proof. Assume that $\beta z \geq \alpha x + \pi$ with $\beta_{jl} = 0$ for all $(j,l) \in A \setminus B$ is not facet defining for P_A. Then all $(x,z) \in P_A$ that satisfy $\beta z = \alpha x + \pi$ also satisfy $\beta^{'} z = \alpha^{'} x + \pi^{'}$ and $(\beta^{'}, \alpha^{'}, \pi^{'}) \neq 0$ is not a positive multiple of (β, α, π). As, for $(j,l) \in A \setminus B$, both (x,z) and $(x,z) + e^z_{jl}$ are in P_A and satisfy $\beta z = \alpha x + \pi$, we have $\beta^{'}_{jl} = 0$. Then $\beta z \geq \alpha x + \pi$ cannot be facet defining for P_B.

If $\beta z \geq \alpha x + \pi$ with $\beta_{jl} = 0$ for all $(j,l) \in A \setminus B$ is facet defining for P_A, then it is clearly facet defining for P_B. □

Theorem 4.7 implies that for $B_1 \subset B_2 \subset A$, facet defining inequalities of P_{B_1} are also facet defining for P_{B_2}. Proposition 4.3 is a special case of Theorem 4.7 where $B = \emptyset$. Facet defining inequalities of $P_\emptyset$ are facet defining for P_B for every $B \subseteq A$.

PROPOSITION 4.8. *For $B \subseteq A$, if $\beta z \geq \alpha x + \pi$ is facet defining for P_B, then $\beta \geq 0$.*

Proof. Let $(x,z) \in P_B$ be such that $\beta z = \alpha x + \pi$. As, for $(j,l) \in B$, $(x,z) + e^z_{jl}$ is also in P_B, $\beta_{jl} \geq 0$. □

Proposition 4.8 implies that facet defining inequalities of P_{jl} that involve both assignment and capacity variables are of the form $z_{jl} \geq \alpha x + \pi$. We give general properties and lifting results for these inequalities.

DEFINITION 4.9. *For $A^{'} \subseteq A$ and $B \subseteq A$, define*

$$F_B(A^{'}) = \left\{(x,z) \in F_B : x_{im} = 0 \; \forall (i,m) \in A \setminus A^{'}\right\}$$

and

$$P_B(A^{'}) = \text{conv}(F_B(A^{'})).$$

If we have a facet defining inequality for $P_B(A^{'})$, then by lifting variables x_{im} with $(i,m) \in A \setminus A^{'}$ sequentially, we can obtain a facet defining inequality for P_B (see, e.g., Nemhauser and Wolsey [13]).

PROPOSITION 4.10. *For $(j,l) \in A$ and $A^{'} \subseteq A$, if $z_{jl} \geq \alpha x + \pi$ is facet defining for $P_{jl}(A^{'})$, then $\alpha_{im} \geq 0$ for $(i,m) \in A^{'}$ such that $i \neq j$ and $i \neq l$.*

Proof. Let $(i,m) \in A^{'}$ such that $i \neq j$ and $i \neq l$. Suppose that $z_{jl} \geq \alpha x + \pi$ is facet defining for $P_{jl}(A^{'})$. Then there exists $(x, z_{jl}) \in P_{jl}(A^{'})$ such that $z_{jl} = \alpha x + \pi$ and $x_{im} = 1$. As $(x, z_{jl}) - e^x_{im}$ is also in $P_{jl}(A^{'})$, we have that $\alpha_{im} \geq 0$. □

The following three theorems give the values of the optimal lifting coefficients of some variables.

THEOREM 4.11. *For $(j,l) \in A$, $A^{'} \subseteq A$ and $(j,u) \in A \setminus A^{'}$, if inequality*

$$z_{jl} \geq \sum_{(i,m) \in A'} \alpha_{im} x_{im} + \pi \tag{4.6}$$

is facet defining for $P_{jl}(A^{'})$, then

$$z_{jl} \geq \sum_{(i,m) \in A'} \alpha_{im} x_{im} + \alpha_{ju} x_{ju} + \pi$$

is facet defining for $P_{jl}(A^{'} \cup \{(j,u)\})$, where

$$\alpha_{ju} = -\max_{x \in F_{\emptyset}(A')} \left(\sum_{(i,m)\in A' : i \neq j, i \neq u \ \ and \ m \neq j} \alpha_{im} x_{im} \right) - \pi.$$

Proof. For x_{ju}, the optimal lifting coefficient α_{ju} can be computed as

$$\alpha_{ju} = \min_{(x,z_{jl}) \in F_{jl}(A' \cup \{(j,u)\}) : x_{ju}=1} \left(z_{jl} - \sum_{(i,m)\in A'} \alpha_{im} x_{im} \right) - \pi.$$

For a given x such that $x_{ju} = 1$, best choice of z_{jl} is 0. So,

$$\alpha_{ju} = \min_{x \in F_{\emptyset}(A' \cup \{(j,u)\}) : x_{ju}=1} \left(- \sum_{(i,m)\in A'} \alpha_{im} x_{im} \right) - \pi.$$

Moreover, as $x_{ju} = 1$, we have $x_{jm} = 0$ for all $m \in I \setminus \{j,u\}$, $x_{ij} = 0$ for all $i \in I \setminus \{j\}$ and $x_{um} = 0$ for all $m \in I \setminus \{u\}$. □

THEOREM 4.12. *For $(j,l) \in A$, $A^{'} \subseteq A$ and $(l,u) \in A \setminus A^{'}$, if inequality* (4.6) *is facet defining for $P_{jl}(A^{'})$, then*

$$z_{jl} \geq \sum_{(i,m)\in A'} \alpha_{im} x_{im} + \alpha_{lu} x_{lu} + \pi$$

is facet defining for $P_{jl}(A^{'} \cup \{(l,u)\})$, where

$$\alpha_{lu} = -\max_{x \in F_{\emptyset}(A')} \left(\sum_{(i,m)\in A' : i \neq l, i \neq u \ \ and \ m \neq l} \alpha_{im} x_{im} \right) - \pi.$$

Proof. The proof is analogous to the proof of Theorem 4.11. □

THEOREM 4.13. *For $(j,l) \in A$ and $A^{'} \subset A$, assume that inequality* (4.6) *is facet defining for $P_{jl}(A^{'})$. Let $(u,v) \in A \setminus A^{'}$ such that u is different from j and l. Consider the two sets of conditions* (i) *and* (ii)*:*

(i) (a) $(j,v) \in A^{'}$,
  (b) *for each $m \in I \setminus \{u,v,j\}$ independently, we have $(u,m) \in A \setminus A^{'}$ or $\alpha_{um} = 0$,*
  (c) *for each $m \in I \setminus \{u,v,j\}$ independently, we have $(m,u) \in A \setminus A^{'}$ or $\alpha_{mu} = 0$.*

(ii) (a) $(l,v) \in A^{'}$,
  (b) *for each $m \in I \setminus \{u,v,l\}$ independently, we have $(u,m) \in A \setminus A^{'}$ or $\alpha_{um} = 0$,*
  (c) *for each $m \in I \setminus \{u,v,l\}$ independently, we have $(m,u) \in A \setminus A^{'}$ or $\alpha_{mu} = 0$.*

If at least one set of conditions (i) *and* (ii) *is satisfied, then inequality* (4.6) *is also facet defining for $P_{jl}(A^{'} \cup \{(u,v)\})$.*

Proof. If inequality (4.6) is facet defining for $P_{jl}(A^{'})$, then inequality

$$z_{jl} \geq \sum_{(i,m)\in A'} \alpha_{im} x_{im} + \alpha_{uv} x_{uv} + \pi$$

is facet defining for $P_{jl}(A^{'} \cup \{(u,v)\})$, where

$$\alpha_{uv} = \min_{(x,z_{jl}) \in P_{jl}(A^{'} \cup \{(u,v)\}): x_{uv}=1} \left(z_{jl} - \sum_{(i,m) \in A'} \alpha_{im} x_{im} \right) - \pi.$$

Assume that condition set (i) is satisfied. As inequality (4.6) is facet defining for $P_{jl}(A^{'})$ and by condition set (i), we know that there exists (x, z_{jl}) in $P_{jl}(A^{'})$ such that $x_{jv} = 1$, $z_{jl} = \sum_{(i,m)\in A'} \alpha_{im} x_{im} + \pi$ and $\sum_{m \in I \setminus \{u,v,j\}} (x_{um} + x_{mu}) = 0$. Then $(x, z_{jl}) + e^{x}_{uv}$ is in $P_{jl}(A^{'} \cup \{(u,v)\})$ and so $\alpha_{uv} \leq 0$. By Proposition 4.10, $\alpha_{uv} \geq 0$. Thus $\alpha_{uv} = 0$. The case where condition set (ii) is satisfied is similar. □

We conclude this section with two more lifting theorems.

Let $(j,l) \in A$, $I_j \subseteq I \setminus \{j,l\}$, $I_l \subseteq I \setminus \{j,l\}$ and $A^{'} = \{(i,j) : i \in I_j\} \cup \{(m,l) : m \in I_l\}$. Consider inequality

$$z_{jl} \geq \sum_{i \in I_j} \alpha_{ij} x_{ij} + \sum_{m \in I_l} \alpha_{ml} x_{ml} + \pi, \tag{4.7}$$

which is facet defining for $P_{jl}(A^{'})$. Let $u \in I \setminus (I_j \cup \{j,l\})$. To compute the lifting coefficient of the variable x_{uj}, we solve a min cut problem on a directed layer graph $G_{uj} = (N_{uj}, A_{uj})$ constructed as follows. Let $I^{'}_j = \{i \in I_j : \alpha_{ij} - t_{il} > 0\}$ and $I^{'}_l = \{m \in I_l : \alpha_{ml} - t_{jm} - t_{um} > 0\}$. Let o and d be two dummy nodes. The node set is $N_{uj} = \{o, d\} \cup I^{'}_j \cup I^{'}_l$. The first layer includes node o, the second layer includes nodes of $I^{'}_j$, the third layer includes nodes of $I^{'}_l$, and the fourth layer includes node d. Arcs go from the nodes of a layer to the nodes of the next layer. Thus, the arc set consists of arcs from node o to nodes in $I^{'}_j$, arcs from nodes in $I^{'}_j$ to nodes in $I^{'}_l$, and arcs from nodes in $I^{'}_l$ to node d, i.e., $A_{uj} = \{(o,i) : i \in I^{'}_j\} \cup \{(i,m) : i \in I^{'}_j, m \in I^{'}_l\} \cup \{(m,d) : m \in I^{'}_l\}$. A cut separating nodes o and d is defined by a subset $C \subset N_{uj}$ with $o \in C$ and $d \notin C$, and the capacity of the cut is the sum of the capacities of arcs going from nodes of C to nodes of $N_{uj} \setminus C$. If there is no such arc, then the cut has zero capacity.

Theorem 4.14. *Let $(j,l) \in A$, $I_j \subseteq I \setminus \{j,l\}$, $I_l \subseteq I \setminus \{j,l\}$, and $A^{'} = \{(i,j) : i \in I_j\} \cup \{(m,l) : m \in I_l\}$. Consider inequality* (4.7) *with integer coefficients.*

Let $u \in I \setminus (I_j \cup \{j,l\})$ and define $I^{'}_j = \{i \in I_j : \alpha_{ij} - t_{il} > 0\}$ and $I^{'}_l = \{m \in I_l : \alpha_{ml} - t_{jm} - t_{um} > 0\}$. Consider the graph $G_{uj} = (N_{uj}, A_{uj})$ constructed above. The capacity of arc $(i,m) \in A_{uj}$ is as follows:

$$w_{im} = \begin{cases} \alpha_{mj} - t_{ml} & \textit{if } i = o \textit{ and } m \in I^{'}_j, \\ \infty & \textit{if } i = m \textit{ and } i \in I^{'}_j \cap I^{'}_l, \\ t_{im} & \textit{if } i \in I^{'}_j \textit{ and } m \in I^{'}_l \setminus \{i\}, \\ \alpha_{il} - t_{ji} - t_{ui} & \textit{if } m = d \textit{ and } i \in I^{'}_l. \end{cases}$$

Let ω be the capacity of a minimum capacity cut separating nodes o and d in the graph $G_{uj} = (N_{uj}, A_{uj})$. Compute

$$\alpha_{uj} = -\pi + \left\lceil t_{jl} + t_{ul} - \sum_{i \in I^{'}_j} (\alpha_{ij} - t_{il}) - \sum_{m \in I^{'}_l} (\alpha_{ml} - t_{jm} - t_{um}) + \omega \right\rceil.$$

If inequality (4.7) *is facet defining for* $P_{jl}(A^{'})$, *then inequality*

$$z_{jl} \geq \sum_{i \in I_j} \alpha_{ij} x_{ij} + \sum_{m \in I_l} \alpha_{ml} x_{ml} + \alpha_{uj} x_{uj} + \pi$$

is facet defining for $P_{jl}(A^{'} \cup \{(u,j)\})$.

Proof. The optimal lifting coefficient of x_{uj} can be computed as follows:

$$\begin{aligned}\alpha_{uj} &= -\pi + \min_{(x,z_{jl}) \in F_{jl}(A^{'} \cup \{(u,j)\}): x_{uj}=1} \left(z_{jl} - \sum_{i \in I_j} \alpha_{ij} x_{ij} - \sum_{m \in I_l} \alpha_{ml} x_{ml} \right) \\ &= -\pi + \min_{(x,z_{jl}) \in F_{jl}(A^{'} \cup \{(u,j)\}): x_{uj}=1} \left[t_{jl} + t_{ul} + \sum_{i \in I_j} (t_{il} - \alpha_{ij}) x_{ij} \right. \\ &\qquad \left. + \sum_{m \in I_l} (t_{jm} + t_{um} - \alpha_{ml}) x_{ml} + \sum_{i \in I_j} \sum_{m \in I_l} t_{im} x_{ij} x_{ml} \right].\end{aligned}$$

There is an optimal solution where $x_{ij} = 0$ for all $i \in I_j \setminus I_j^{'}$ and $x_{ml} = 0$ for all $m \in I_l \setminus I_l^{'}$. So,

$$\begin{aligned}\alpha_{uj} &= -\pi + \left[t_{jl} + t_{ul} + \min_{(x,z_{jl}) \in F_{jl}(A^{'} \cup \{(u,j)\}): x_{uj}=1} \left(\sum_{i \in I_j^{'}} (t_{il} - \alpha_{ij}) x_{ij} \right. \right. \\ &\qquad \left. \left. + \sum_{m \in I_l^{'}} (t_{jm} + t_{um} - \alpha_{ml}) x_{ml} + \sum_{i \in I_j^{'}} \sum_{m \in I_l^{'}} t_{im} x_{ij} x_{ml} \right) \right] \\ &= -\pi + \left[t_{jl} + t_{ul} - \sum_{i \in I_j^{'}} (\alpha_{ij} - t_{il}) - \sum_{m \in I_l^{'}} (\alpha_{ml} - t_{jm} - t_{um}) \right. \\ &\qquad + \min_{(x,z_{jl}) \in F_{jl}(A^{'} \cup \{(u,j)\}): x_{uj}=1} \left(\sum_{i \in I_j^{'}} (\alpha_{ij} - t_{il})(1 - x_{ij}) \right. \\ &\qquad \left. \left. + \sum_{m \in I_l^{'}} (\alpha_{ml} - t_{jm} - t_{um})(1 - x_{ml}) + \sum_{i \in I_j^{'}} \sum_{m \in I_l^{'}} t_{im} x_{ij} x_{ml} \right) \right].\end{aligned}$$

It remains to show that ω is equal to the optimal value of the above minimization problem. Let C be a cut separating nodes o and d in G_{uj}. The capacity of cut C is

$$\sum_{i \in I_j^{'} \setminus C} (\alpha_{ij} - t_{il}) + \sum_{m \in I_l^{'} \cap C} (\alpha_{ml} - t_{jm} - t_{um}) + \sum_{i \in I_j^{'} \cap C} \sum_{m \in I_l^{'} \setminus C} t_{im}.$$

This is the cost of a solution where x_{ij} is equal to 1 if $i \in I_j^{'} \cap C$ and 0 otherwise for $i \in I_j^{'}$ and x_{ml} is equal to 1 if $m \in I_l^{'} \setminus C$ and 0 otherwise for $m \in I_l^{'}$. The solution is infeasible if there exists $i \in I_j^{'} \cap I_l^{'}$ such that $x_{ij} + x_{il} = 2$. Then the corresponding cut has infinite capacity since $w_{ii} = \infty$ for all $i \in I_j^{'} \cap I_l^{'}$. Therefore, any feasible solution of the minimization problem is a cut with a finite capacity and vice versa. Besides, the cost of a feasible solution is the same as the capacity of the corresponding cut. So ω is the same as the optimal value of the minimization problem. □

THEOREM 4.15. *Let $(j,l) \in A$, $I_j \subseteq I \setminus \{j,l\}$, $I_l \subseteq I \setminus \{j,l\}$, and $A^{'} = \{(i,j) : i \in I_j\} \cup \{(m,l) : m \in I_l\}$. Consider inequality* (4.7) *with integer coefficients.*

Let $u \in I \setminus (I_l \cup \{j,l\})$ and define $I_j^{'} = \{i \in I_j : \alpha_{ij} - t_{il} - t_{iu} > 0\}$ and $I_l^{'} = \{m \in I_l : \alpha_{ml} - t_{jm} > 0\}$. Consider the graph $G_{ul} = (N_{ul}, A_{ul})$. The node set is $N_{ul} = \{o,d\} \cup I_j^{'} \cup I_l^{'}$, and nodes o and d are dummy nodes. The arc set is $A_{ul} = \{(o,i) : i \in I_j^{'}\} \cup \{(i,m) : i \in I_j^{'}, m \in I_l^{'}\} \cup \{(m,d) : m \in I_l^{'}\}$. The capacity of arc $(i,m) \in A_{ul}$ is as follows:

$$w_{im} = \begin{cases} \alpha_{mj} - t_{ml} - t_{mu} & \text{if } i = o \text{ and } m \in I_j^{'}, \\ \infty & \text{if } i = m \text{ and } i \in I_j^{'} \cap I_l^{'}, \\ t_{im} & \text{if } i \in I_j^{'} \text{ and } m \in I_l^{'} \setminus \{i\}, \\ \alpha_{il} - t_{ji} & \text{if } m = d \text{ and } i \in I_l^{'}. \end{cases}$$

Let ω be the capacity of a minimum capacity cut separating nodes o and d in the graph $G_{ul} = (N_{ul}, A_{ul})$. Compute

$$\alpha_{ul} = -\pi + \left\lceil t_{jl} + t_{ju} - \sum_{i \in I_j^{'}} (\alpha_{ij} - t_{il} - t_{iu}) - \sum_{m \in I_l^{'}} (\alpha_{ml} - t_{jm}) + \omega \right\rceil.$$

If inequality (4.7) *is facet defining for $P_{jl}(A^{'})$, then inequality*

$$z_{jl} \geq \sum_{i \in I_j} \alpha_{ij} x_{ij} + \sum_{m \in I_l} \alpha_{ml} x_{ml} + \alpha_{ul} x_{ul} + \pi$$

is facet defining for $P_{jl}(A^{'} \cup \{(u,l)\})$.

4.3. Facets of P_{jl}. We present families of facet defining inequalities of P_{jl} for $(j,l) \in A$. By Theorem 4.7, these inequalities are also facet defining for P_A.

We use sequential lifting to derive facet defining inequalities for P_{jl}. We start with the inequality $z_{jl} \geq \lceil t_{jl} \rceil$, which is facet defining for $P_{jl}(\emptyset)$. For a subset $I^{'} \subseteq I \setminus \{j,l\}$ and an order ϕ on $I^{'}$, we first lift the variables x_{ij} for $i \in I^{'}$ in the order ϕ. The remaining variables are lifted in the following order: x_{jm} for $m \in I \setminus \{j\}$, x_{uv} with $u \in I \setminus \{j,l\}$ and $v \in I \setminus \{j,u\}$, x_{lm} with $m \in I \setminus \{l\}$, and x_{uj} with $u \in I \setminus (I^{'} \cup \{j,l\})$. As all lifting coefficients are optimal, the resulting inequality is facet defining for P_{jl}.

THEOREM 4.16. *Let $(j,l) \in A$, $I^{'} \subseteq I \setminus \{j,l\}$ and ϕ be an order on $I^{'}$. For $i \in I^{'}$,*

$$\alpha_{ij} = -\lceil t_{jl} \rceil + \left\lceil t_{jl} + t_{il} - \sum_{m \in I^{'} : \phi(m) < \phi(i)} (\alpha_{mj} - t_{ml})^{+} \right\rceil.$$

Inequality

$$z_{jl} \geq \lceil t_{jl} \rceil \left(1 - \sum_{m \in I \setminus \{j\}} x_{jm} - \sum_{m \in I \setminus \{l\}} x_{lm} \right) + \sum_{i \in I^{'}} \alpha_{ij} \left(x_{ij} - \sum_{m \in I \setminus \{l,i\}} x_{lm} \right) \tag{4.8}$$

is facet defining for P_{jl}.

Proof. Inequality $z_{jl} \geq \lceil t_{jl} \rceil$ is facet defining for $P_{jl}(\emptyset)$. We lift variables x_{ij} for $i \in I^{'}$ in the order ϕ. Let

$$F_{jl}^{i} = \left\{ (x, z_{jl}) \in F_{jl}\Big(\big\{ (m,j) \in A : \phi(m) \leq \phi(i) \big\} \Big) : x_{ij} = 1 \right\}.$$

The optimal lifting coefficient for x_{ij} is

$$\alpha_{ij} = \min_{(x,z_{jl})\in F^i_{jl}} \left(z_{jl} - \lceil t_{jl} \rceil - \sum_{m\in I^{'}:\phi(m)<\phi(i)} \alpha_{mj} x_{mj} \right).$$

For x such that $x_{ij} = 1$, the lowest value of z_{jl} is

$$\left\lceil t_{jl} + t_{il} + \sum_{m\in I^{'}:\phi(m)<\phi(i)} t_{ml} x_{mj} \right\rceil.$$

Thus,

$$\alpha_{ij} = \min_{(x,z_{jl})\in F^i_{jl}} \left(\left\lceil t_{jl} + t_{il} + \sum_{m\in I^{'}:\phi(m)<\phi(i)} t_{ml} x_{mj} \right\rceil - \sum_{m\in I^{'}:\phi(m)<\phi(i)} \alpha_{mj} x_{mj} \right) - \lceil t_{jl} \rceil.$$

By induction, one can show that α_{mj} is an integer for each $m \in I^{'}$ such that $\phi(m) < \phi(i)$. So,

$$\alpha_{ij} = -\lceil t_{jl} \rceil + \min_{(x,z_{jl})\in F^i_{jl}} \left\lceil t_{jl} + t_{il} + \sum_{m\in I^{'}:\phi(m)<\phi(i)} (t_{ml} - \alpha_{mj}) x_{mj} \right\rceil.$$

The minimization problem can be solved by setting $x_{mj} = 1$ for $m \in I^{'}$ with $\phi(m) < \phi(i)$ if $\alpha_{mj} - t_{ml} \geq 0$ and at 0 otherwise.

Next we lift variables x_{jm}. For $m \in I \setminus \{j\}$, Theorem 4.11 implies that $\alpha_{jm} = -\lceil t_{jl} \rceil$.

Now consider some x_{uv} with $u \in I \setminus \{j, l\}$ and $v \in I \setminus \{j, u\}$. We prove by induction that $\alpha_{uv} = 0$. If x_{uv} is the first variable with $u \in I \setminus \{j, l\}$ and $v \in I \setminus \{j, u\}$ to lift, then as x_{jv} is already lifted and for each $m \in I \setminus \{u, v, j\}$, x_{um} and x_{mu} are not yet lifted, condition set (i) of Theorem 4.13 is satisfied and the lifting coefficient of x_{uv} is zero. Otherwise, assume that those x_{im} with $i \in I \setminus \{j, l\}$ and $m \in I \setminus \{j, i\}$ that are already lifted have zero coefficient. Then as x_{jv} is already lifted and for each $m \in I \setminus \{u, v, j\}$, x_{um} is not lifted or it has zero lifting coefficient and x_{mu} is not lifted or it has zero lifting coefficient, condition set (i) of Theorem 4.13 is satisfied. Hence, the lifting coefficient of x_{uv} is zero.

We lift variables x_{lm}. For $m \in I \setminus \{l\}$, as by Proposition 4.10 $\alpha_{ij} \geq 0$ for all $i \in I^{'}$ and $\alpha_{ji} \leq 0$ for all $i \in I \setminus \{j\}$, Theorem 4.12 implies that $\alpha_{lm} = -\sum_{i\in I^{'}\setminus\{m\}} \alpha_{ij} - \lceil t_{jl} \rceil$.

Finally variables x_{uj} with $u \in I \setminus (I^{'} \cup \{j, l\})$ are lifted by applying Theorem 4.13 repeatedly. As x_{lj} is already lifted and for each $m \in I \setminus \{u, j, l\}$, the lifting coefficients of x_{um} and x_{mu} are zero, condition set (ii) of Theorem 4.13 is satisfied and the lifting coefficient of x_{uj} is zero. □

The three corollaries below present facet defining inequalities that are special cases of inequalities (4.8) for $|I^{'}| \leq 2$.

Corollary 4.17. *For $(j, l) \in A$, inequality*

$$z_{jl} \geq \lceil t_{jl} \rceil \left(1 - \sum_{m\in I\setminus\{j\}} x_{jm} - \sum_{m\in I\setminus\{l\}} x_{lm} \right) \tag{4.9}$$

is facet defining for P_{jl}.

COROLLARY 4.18. *For $(j,l) \in A$ and $u \in I \setminus \{j,l\}$, inequality*

$$\begin{aligned} z_{jl} \geq \lceil t_{jl} \rceil \Bigg(1 - \sum_{m \in I\setminus\{j\}} x_{jm} - \sum_{m\in I\setminus\{l\}} x_{lm}\Bigg) \\ + \Big(\lceil t_{jl} + t_{ul} \rceil - \lceil t_{jl} \rceil\Big)\Bigg(x_{uj} - \sum_{m \in I\setminus\{l,u\}} x_{lm}\Bigg) \end{aligned} \tag{4.10}$$

is facet defining for P_{jl}.

COROLLARY 4.19. *Let $(j,l) \in A$ and $u, v \in I \setminus \{j,l\}$ such that $u \neq v$. Let $a = \min\{\lceil t_{jl} + t_{ul} + t_{vl}\rceil - \lceil t_{jl} + t_{ul}\rceil, \lceil t_{jl} + t_{vl}\rceil - \lceil t_{jl}\rceil\}$. Inequality*

$$\begin{aligned} z_{jl} \geq \lceil t_{jl} \rceil \Bigg(1 - \sum_{m \in I\setminus\{j\}} x_{jm} - \sum_{m\in I\setminus\{l\}} x_{lm}\Bigg) + a\Bigg(x_{vj} - \sum_{m \in I\setminus\{l,v\}} x_{lm}\Bigg) \\ + \Big(\lceil t_{jl} + t_{ul} \rceil - \lceil t_{jl} \rceil\Big)\Bigg(x_{uj} - \sum_{m \in I\setminus\{l,u\}} x_{lm}\Bigg) \end{aligned} \tag{4.11}$$

is facet defining for P_{jl}.

THEOREM 4.20. *Let $(j,l) \in A$, $I^{'} \subseteq I\setminus\{j,l\}$ and ϕ be an order on $I^{'}$. For $i \in I^{'}$,*

$$\alpha_{il} = -\lceil t_{jl} \rceil + \left\lceil t_{jl} + t_{ji} - \sum_{m\in I^{'}:\phi(m)<\phi(i)} (\alpha_{ml} - t_{jm})^{+} \right\rceil.$$

Inequality

$$z_{jl} \geq \lceil t_{jl} \rceil \Bigg(1 - \sum_{m \in I\setminus\{j\}} x_{jm} - \sum_{m\in I\setminus\{l\}} x_{lm}\Bigg) + \sum_{i\in I^{'}} \alpha_{il}\Bigg(x_{il} - \sum_{m\in I\setminus\{j,i\}} x_{jm}\Bigg) \tag{4.12}$$

is facet defining for P_{jl}.

Proof. Analogous to the proof of Theorem 4.16. □

Facet defining inequalities can also be obtained by fixing the values of some variables to 1 and applying sequential lifting.

Let A_0 and A_1 be disjoint subsets of A. For $(j,l) \in A$, define

$$\overline{F}_{jl}(A_0, A_1) = F_{jl} \cap \big\{(x, z_{jl}) : x_{im} = 0 \ \forall (i,m) \in A_0 \text{ and } x_{im} = 1 \ \forall (i,m) \in A_1\big\}$$

and

$$\overline{P}_{jl}(A_0, A_1) = \operatorname{conv}\big(\overline{F}_{jl}(A_0, A_1)\big).$$

Let $I^{'} \subseteq I\setminus\{j,l\}$ and $A_1 = \{(i,j) \in A : i \in I^{'}\}$. Inequality $z_{jl} \geq \lceil \sum_{m\in I^{'}} t_{ml} + t_{jl}\rceil$ is facet defining for $\overline{P}_{jl}(A \setminus A_1, A_1)$. To derive a facet defining inequality for P_{jl}, we first lift $(1 - x_{ij})$ for $i \in I^{'}$ in some order ϕ, then x_{jm} for $m \in I \setminus \{j\}$, x_{uv} with $u \in I \setminus \{j,l\}$ and $v \in I \setminus \{j,u\}$, x_{lm} with $m \in I \setminus \{l\}$, and finally x_{uj} with $u \in I \setminus (I^{'} \cup \{j,l\})$.

THEOREM 4.21. *Let $(j,l) \in A$, $I^{'} \subseteq I\setminus\{j,l\}$ and ϕ be an order on $I^{'}$. For $i \in I^{'}$,*

$$\alpha_{ij} = -\left\lceil \sum_{m\in I^{'}} t_{ml} + t_{jl} \right\rceil + \left\lceil t_{jl} + \sum_{m\in I^{'}\setminus\{i\}} t_{ml} - \sum_{m\in I^{'}:\phi(m)<\phi(i)} (t_{ml} + \alpha_{mj})^{+} \right\rceil.$$

Inequality

$$\begin{aligned} z_{jl} \geq & \sum_{i\in I'} \alpha_{ij}\Big(1 - x_{ij} - x_{li} - \sum_{m\in I\setminus\{j\}} x_{jm}\Big) \\ & + \Big\lceil \sum_{m\in I'} t_{ml} + t_{jl}\Big\rceil \Big(1 - \sum_{m\in I\setminus\{j\}} x_{jm} - \sum_{m\in I\setminus\{l\}} x_{lm}\Big) \end{aligned} \tag{4.13}$$

is facet defining for P_{jl}.

Proof. Let $A_1 = \{(i,j) \in A : i \in I^{'}\}$. Inequality $z_{jl} \geq \lceil \sum_{m\in I'} t_{ml} + t_{jl}\rceil$ is facet defining for $\overline{P}_{jl}(A \setminus A_1, A_1)$. We lift $(1 - x_{ij})$ for $i \in I^{'}$ in the order ϕ. Let

$$F^i_{jl} = \overline{F}_{jl}\big(A \setminus A_1 \cup \{(i,j)\}, A_1 \setminus \{(m,j) : \phi(m) \leq \phi(i)\}\big).$$

The optimal lifting coefficient for $(1 - x_{ij})$ is

$$\alpha_{ij} = \min_{(x,z_{jl})\in F^i_{jl}} \Bigg(z_{jl} - \sum_{m\in I':\phi(m)<\phi(i)} \alpha_{mj}(1 - x_{mj})\Bigg) - \Bigg\lceil \sum_{m\in I'} t_{ml} + t_{jl}\Bigg\rceil.$$

For x such that $x_{ij} = 0$, the lowest value for z_{jl} is

$$z_{jl} = \Bigg\lceil t_{jl} + \sum_{m\in I':\phi(m)>\phi(i)} t_{ml} + \sum_{m\in I':\phi(m)<\phi(i)} t_{ml}x_{mj}\Bigg\rceil.$$

Then

$$\begin{aligned} \alpha_{ij} = \min_{(x,z_{jl})\in F^i_{jl}} & \Bigg(\Bigg\lceil t_{jl} + \sum_{m\in I':\phi(m)>\phi(i)} t_{ml} + \sum_{m\in I':\phi(m)<\phi(i)} t_{ml}x_{mj}\Bigg\rceil \\ & - \sum_{m\in I':\phi(m)<\phi(i)} \alpha_{mj}(1 - x_{mj})\Bigg) - \Bigg\lceil \sum_{m\in I'} t_{ml} + t_{jl}\Bigg\rceil. \end{aligned}$$

By induction, one can again show that α_{mj} is integer for each $m \in I^{'}$ such that $\phi(m) < \phi(i)$. So

$$\begin{aligned} \alpha_{ij} &= -\Bigg\lceil \sum_{m\in I'} t_{ml} + t_{jl}\Bigg\rceil + \min_{(x,z_{jl})\in F^i_{jl}} \Bigg\lceil t_{jl} + \sum_{m\in I'\setminus\{i\}} t_{ml} \\ &\qquad\qquad - \sum_{m\in I':\phi(m)<\phi(i)} (t_{ml} + \alpha_{mj})(1 - x_{mj})\Bigg\rceil \\ &= -\Bigg\lceil \sum_{m\in I'} t_{ml} + t_{jl}\Bigg\rceil + \Bigg\lceil t_{jl} + \sum_{m\in I'\setminus\{i\}} t_{ml} - \sum_{m\in I':\phi(m)<\phi(i)} (t_{ml} + \alpha_{mj})^+\Bigg\rceil. \end{aligned}$$

Next, we lift variables x_{jm}. For $m \in I \setminus \{j\}$, $\alpha_{jm} = -\sum_{i\in I'} \alpha_{ij} - \lceil \sum_{i\in I'} t_{il} + t_{jl}\rceil$ since $x_{ij} = 0$ for all $i \in I^{'}$ as $x_{jm} = 1$.

Now we lift variables x_{uv} with $u \in I \setminus \{j, l\}$ and $v \in I \setminus \{j, u\}$. As condition set (i) of Theorem 4.13 is satisfied, these variables have zero lifting coefficient (see the proof of Theorem 4.16).

Next, we lift variables x_{lm}. Let $m \in I \setminus \{j,l\}$. Since $\alpha_{ij} \leq 0$ for all $i \in I^{'}$ and α_{ji} is the same for all $i \in I \setminus \{j\}$, by Theorem 4.12, the optimal lifting coefficient for x_{lm} is

$$\alpha_{lm} = -\sum_{i \in I'} \alpha_{ij} - \left\lceil \sum_{i \in I'} t_{il} + t_{jl} \right\rceil + \min\left\{ \sum_{i \in I' \setminus \{m\}} \alpha_{ij}, \sum_{i \in I'} \alpha_{ij} + \left\lceil \sum_{i \in I'} t_{il} + t_{jl} \right\rceil \right\}.$$

If $m \notin I^{'}$, then $\alpha_{lm} = -\lceil \sum_{i \in I'} t_{il} + t_{jl} \rceil$. If $m \in I^{'}$, then

$$\alpha_{lm} = -\sum_{i \in I'} \alpha_{ij} - \left\lceil \sum_{i \in I'} t_{il} + t_{jl} \right\rceil + \min\left\{ \sum_{i \in I'} \alpha_{ij} - \alpha_{mj}, \sum_{i \in I'} \alpha_{ij} + \left\lceil \sum_{i \in I'} t_{il} + t_{jl} \right\rceil \right\}.$$

This is the same as $\min\{-\alpha_{mj} - \lceil \sum_{i \in I'} t_{il} + t_{jl} \rceil, 0\}$. As

$$\left\lceil t_{jl} + \sum_{i \in I' \setminus \{m\}} t_{il} - \sum_{i \in I' : \phi(i) < \phi(m)} (t_{il} + \alpha_{ij})^{+} \right\rceil \geq 0,$$

we get $\alpha_{lm} = -\alpha_{mj} - \lceil \sum_{i \in I'} t_{il} + t_{jl} \rceil$.

We lift x_{lj}. As $\sum_{i \in I \setminus \{j\}} x_{ji} = 0$, $\alpha_{lj} = -\lceil \sum_{i \in I'} t_{il} + t_{jl} \rceil$.

Finally variables x_{uj} with $u \in I \setminus (I^{'} \cup \{j,l\})$ are lifted by applying Theorem 4.13 repeatedly and their lifting coefficients are zero (see proof of Theorem 4.16).

The resulting inequality is

$$z_{jl} \geq \left\lceil \sum_{m \in I'} t_{ml} + t_{jl} \right\rceil + \sum_{i \in I'} \alpha_{ij}(1 - x_{ij}) - \sum_{m \in I \setminus \{j\}} \left(\sum_{i \in I'} \alpha_{ij} + \left\lceil \sum_{i \in I'} t_{il} + t_{jl} \right\rceil \right) x_{jm}$$
$$- \sum_{m \in I \setminus (I' \cup \{l\})} \left\lceil \sum_{i \in I'} t_{il} + t_{jl} \right\rceil x_{lm} - \sum_{m \in I'} \left(\alpha_{mj} + \left\lceil \sum_{i \in I'} t_{il} + t_{jl} \right\rceil \right) x_{lm}.$$

Rearranging terms, we obtain inequality (4.13). □

For $I^{'} = \emptyset$ and $I^{'} = \{u\}$, inequality (4.13) reduces to inequalities (4.9) and (4.10), respectively. Inequality (4.13) for $I^{'} = \{u, v\}$, $\phi(u) = 1$ and $\phi(v) = 2$ is given in the following corollary.

COROLLARY 4.22. *Let $(j,l) \in A$ and $u, v \in I \setminus \{j,l\}$ such that $u \neq v$. Let $a = \max\{\lceil t_{jl} + t_{ul} + t_{vl} \rceil - \lceil t_{jl} + t_{ul} \rceil, \lceil t_{jl} + t_{vl} \rceil - \lceil t_{jl} \rceil\}$. Inequality*

$$z_{jl} \geq \Big(\lceil t_{jl} + t_{vl} \rceil - a \Big) \Big(1 - \sum_{m \in I \setminus \{j\}} x_{jm} \Big) - \lceil t_{jl} + t_{ul} + t_{vl} \rceil \sum_{m \in I \setminus \{l\}} x_{lm}$$
$$+ \Big(\lceil t_{jl} + t_{ul} + t_{vl} \rceil - \lceil t_{jl} + t_{vl} \rceil \Big)(x_{uj} + x_{lu}) + a(x_{vj} + x_{lv}) \tag{4.14}$$

is facet defining for P_{jl}.

THEOREM 4.23. *Let $(j,l) \in A$, $I^{'} \subseteq I \setminus \{j,l\}$ and ϕ be an order on $I^{'}$. For $i \in I^{'}$,*

$$\alpha_{il} = -\left\lceil \sum_{m \in I'} t_{jm} + t_{jl} \right\rceil + \left\lceil t_{jl} + \sum_{m \in I' \setminus \{i\}} t_{jm} - \sum_{m \in I' : \phi(m) < \phi(i)} (t_{jm} + \alpha_{ml})^{+} \right\rceil.$$

Inequality

$$z_{jl} \geq \sum_{i\in I'} \alpha_{il}\Big(1 - x_{il} - x_{ji} - \sum_{m\in I\setminus\{l\}} x_{lm}\Big) \tag{4.15}$$
$$+ \Big\lceil \sum_{m\in I'} t_{jm} + t_{jl}\Big\rceil \Big(1 - \sum_{m\in I\setminus\{j\}} x_{jm} - \sum_{m\in I\setminus\{l\}} x_{lm}\Big)$$

is facet defining for P_{jl}.

Proof. The proof is analogous to the proof of Theorem 4.21. □

Finally, using Theorems 4.14 and 4.15, we find the following facet defining inequalities.

PROPOSITION 4.24. *Let $(j,l) \in A$ and $u, v \in I \setminus \{j,l\}$ such that $u \neq v$. Let $a = \min\{\lceil t_{jl} + t_{jv}\rceil - \lceil t_{jl}\rceil, \lceil t_{jl} + t_{jv} + t_{ul} + t_{uv}\rceil - \lceil t_{jl} + t_{ul}\rceil\}$. Inequality*

$$z_{jl} \geq \lceil t_{jl}\rceil \Big(1 - \sum_{m\in I\setminus\{j\}} x_{jm} - \sum_{m\in I\setminus\{l\}} x_{lm}\Big) + a\Big(x_{vl} - \sum_{m\in I\setminus\{j,v\}} x_{jm}\Big)$$
$$+\Big(\lceil t_{jl} + t_{ul}\rceil - \lceil t_{jl}\rceil\Big)\Big(x_{uj} - \sum_{m\in I\setminus\{l,u\}} x_{lm}\Big) \tag{4.16}$$

is facet defining for P_{jl}.

Proof. Inequality $z_{jl} \geq \lceil t_{jl}\rceil$ is facet defining for $P_{jl}(\emptyset)$. Now lift first x_{uj} and then x_{vl} using Theorems 4.14 and 4.15, respectively. Inequality

$$z_{jl} \geq \lceil t_{jl}\rceil + (\lceil t_{jl} + t_{ul}\rceil - \lceil t_{jl}\rceil)x_{uj} + a x_{vl}$$

is facet defining for $P_{jl}(\{(u,j),(v,l)\})$. Next, by Theorem 4.11, optimal lifting coefficient for x_{jm} with $m \in I \setminus \{j,v\}$ is $-\lceil t_{jl}\rceil - a$ and for x_{jv} is $-\lceil t_{jl}\rceil$. The optimal coefficient of x_{lm} for $m \in I \setminus \{l,u\}$ is $-\lceil t_{jl} + t_{ul}\rceil$ and for x_{lu} is $-\lceil t_{jl}\rceil$.

Next, we lift variables x_{ij} with $i \in I \setminus \{u,l,j\}$. As x_{lj} is already lifted and for $m \in I \setminus \{i,j,l\}$, x_{im} and x_{mi} are not lifted, condition set (ii) of Theorem 4.13 is satisfied and the lifting coefficient of x_{ij} is zero.

For x_{il} with $i \in I\setminus\{v,j,l\}$, as x_{jl} is already lifted and for $m \in I\setminus\{i,j,l\}$, x_{im} and x_{mi} are not lifted, condition set (i) of Theorem 4.13 is satisfied. So lifting coefficient of x_{il} is zero.

Inequality (4.16) is facet defining for $P_{jl}(A')$, where $A' = A \setminus \{(i,k) : i \in I \setminus \{j,l\}, k \in I \setminus \{i,j,l\}\}$. Next we lift x_{ik} with $i \in I\setminus\{j,l\}$ and $k \in I\setminus\{i,j,l\}$. Optimal lifting coefficient is

$$\alpha_{ik} = \min_{(x,z_{jl})\in F_{jl}(A'\cup\{(i,k)\}):x_{ik}=1} \sigma(x, z_{jl}),$$

where

$$\sigma(x,z_{jl}) = \Bigg(z_{jl} - \lceil t_{jl}\rceil \Big(1 - \sum_{m\in I\setminus\{j\}} x_{jm} - \sum_{m\in I\setminus\{l\}} x_{lm}\Big) - a\Big(x_{vl} - \sum_{m\in I\setminus\{j,v\}} x_{jm}\Big)$$
$$-\Big(\lceil t_{jl} + t_{ul}\rceil - \lceil t_{jl}\rceil\Big)\Big(x_{uj} - \sum_{m\in I\setminus\{l,u\}} x_{lm}\Big)\Bigg).$$

If $i \neq v$ and $k \neq v$, then let $x = e^x_{ik} + e^x_{jk} + e^x_{vl}$ and $z_{jl} = 0$. If $i \neq v$ and $k = v$, then let $x = e^x_{iv} + e^x_{jv}$ and $z_{jl} = 0$. If $i = v$ and $k \neq u$, then $x = e^x_{vk} + e^x_{lk} + e^x_{uj}$ and $z_{jl} = 0$. Finally, if $i = v$ and $k = u$, then $x = e^x_{vu} + e^x_{lu}$ and $z_{jl} = 0$. Solution $(x, z_{jl}) \in F_{jl}(A^{'} \cup \{(i,k)\})$ with $x_{ik} = 1$ and $\sigma(x, z_{jl}) = 0$. We know by Proposition 4.10 that $\alpha_{ik} \geq 0$. Therefore, $\alpha_{ik} = 0$. Repeating the same argument, we can prove that lifting coefficients of all variables x_{ik} with $i \in I \setminus \{j,l\}$ and $k \in I \setminus \{i,j,l\}$ are zero. □

An important issue is the separation of these inequalities. Inequalities (4.9), (4.10), (4.11), (4.14), and (4.16) can be separated in polynomial time by enumeration. The separation of inequalities (4.8), (4.12), (4.13), and (4.15) asks to choose a subset $I^{'} \subseteq I \setminus \{j,l\}$ and to find an order ϕ on $I^{'}$. We do not know the complexity of these problems.

5. Conclusion. In this paper, we presented polyhedral results for the HLM. By previous results, it was easy to characterize the facet defining inequalities that involve only the assignment or the capacity variables. It remained to investigate strong valid inequalities that involved both types of variables. We presented valid inequalities, results that give the optimal lifting coefficients of some variables as well as families of facet defining inequalities.

A future research direction is to study similar lifting results for P_B where $B \subseteq A$ is not necessarily a singleton. Another one is to find efficient separation algorithms for the inequalities given here and incorporate these results in a branch and cut algorithm.

Acknowledgments. The author is grateful to an anonymous referee for his or her helpful comments on the structure and presentation and for drawing attention to several errors.

REFERENCES

[1] P. Avella and A. Sassano, *On the p-median polytope*, Math. Program., 89 (2001), pp. 395–411.

[2] E. Balas and M. W. Padberg, *Set partitioning: A survey*, SIAM Rev., 18 (1976), pp. 710–760.

[3] J. F. Campbell, A. T. Ernst, and M. Krishnamoorthy, *Hub location problems*, in Facility Location: Applications and Theory, Z. Drezner and H. W. Hamacher, eds., Springer, New York, 2002, pp. 373–407.

[4] L. Cánovas, M. Landete, and A. Marín, *Facet obtaining procedures for set packing problems*, SIAM J. Discrete Math., 16 (2003), pp. 127–155.

[5] S. Chung, Y. Myung, and D. Tcha, *Optimal design of a distributed network with a two-level hierarchical structure*, European J. Oper. Res., 62 (1992), pp. 105–115.

[6] H. W. Hamacher, M. Labbé, S. Nickel, and T. Sonneborn, *Adapting polyhedral properties from facility to hub location problems*, Discrete Appl. Math., 145 (2004), pp. 104–116.

[7] J. G. Klincewicz, *Hub location in backbone/tributary network design: A review*, Location Sci., 6 (1998), pp. 307–335.

[8] M. Labbé and H. Yaman, *A Note on the Projection of Polyhedra*, Preprint 2003/18, Université Libre de Bruxelles, 2003; also available online from http://www.ulb.ac.be/di/gom/publications/technical/2003.html.

[9] M. Labbé and H. Yaman, *Polyhedral Analysis for Concentrator Location Problems*, Preprint 2003/13, Université Libre de Bruxelles, 2003; also available online from http://www.ulb.ac.be/di/gom/publications/technical/2003.html.

[10] M. Labbé and H. Yaman, *Projecting the flow variables for hub location problems*, Networks, 44 (2004), pp. 84–93.

[11] M. Labbé and H. Yaman, *Solving the Uncapacitated Concentrator Location Problem with Star Routing*, Preprint 2003/15, Université Libre de Bruxelles, 2003; also available online from http://www.ulb.ac.be/di/gom/publications/technical/2003.html.

[12] M. Labbé, H. Yaman, and E. Gourdin, *A branch and cut algorithm for hub location problems with single assignment*, in Math. Program., 102 (2005), pp. 371–405.

[13] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley, New York, 1988.

[14] M. W. Padberg, *On the facial structure of set packing polyhedra*, Math. Program., 5 (1973), pp. 199–215.

[15] H. Yaman and G. Carello, *Solving the hub location problem with modular link capacities*, Comput. Oper. Res., 32 (2005), pp. 3227–3245.

SIAM J. Discrete Math.
Vol. 19, No. 2, pp. 523–542

TRAFFIC GROOMING IN UNIDIRECTIONAL WAVELENGTH-DIVISION MULTIPLEXED RINGS WITH GROOMING RATIO $C = 6$*

JEAN-CLAUDE BERMOND†, CHARLES J. COLBOURN‡, DAVID COUDERT†, GENNIAN GE§, ALAN C. H. LING¶, AND XAVIER MUÑOZ‖

Abstract. SONET/WDM networks using wavelength add-drop multiplexing can be constructed using certain graph decompositions used to form a grooming, consisting of unions of primitive rings. The cost of such a decomposition is the sum, over all graphs in the decomposition, of the number of vertices of nonzero degree in the graph. The existence of such decompositions with minimum cost, when every pair of sites employs no more than $\frac{1}{6}$ of the wavelength capacity, is determined with a finite number of possible exceptions. Indeed, when the number N of sites satisfies $N \equiv 1 \pmod 3$, the determination is complete, and when $N \equiv 2 \pmod 3$, the only value left undetermined is $N = 17$. When $N \equiv 0 \pmod 3$, a finite number of values of N remain, the largest being $N = 2580$. The techniques developed rely heavily on tools from combinatorial design theory.

Key words. traffic grooming, combinatorial designs, block designs, group-divisible designs, optical networks, wavelength-division multiplexing

AMS subject classifications. 68M10, 68R05

DOI. 10.1137/S0895480104444314

1. Traffic grooming in wavelength-division multiplexed rings.

Many current network infrastructures are based on the synchronous optical network (SONET). A SONET ring typically consists of a set of nodes connected by an optical fiber in a unidirectional ring topology. Nodes of the network insert and/or extract the data streams on a wavelength by means of an add drop multiplexer (ADM). A *wavelength-division multiplexed* (WDM) or *dense WDM* (DWDM) optical network can handle many wavelengths, each with large bandwidth available. On the other hand, a single user seldom needs such large bandwidth. Therefore, by using multiplexed access such as time-division multiple access (TDMA) or code-division multiple access (CDMA), different users can share the same wavelength, thereby optimizing the bandwidth usage of the network. Traffic grooming is the generic term for packing low rate signals into higher speed streams (see [17, 32, 34]). By using traffic grooming, not only is the bandwidth usage optimized, but also the cost of the network can be reduced by lessening the total number of ADMs. If traffic grooming is used, one node may or may not

*Received by the editors June 5, 2004; accepted for publication (in revised form) January 28, 2005; published electronically November 4, 2005. This work has been partially funded by European projects RTN ARACNE and FET CRESCCO, by the U.S. Army Research Office under grant DAAD19-01-1-0406, by the National Natural Science Foundation of China under grant 10471127, by the Zhejiang Provincial Natural Science Foundation of China under grant R604001, and by the Spanish government under projects TIC-2001-2171 and TIC-2002-00155.

http://www.siam.org/journals/sidma/19-2/44431.html

†Mascotte Project, I3S-CNRS, INRIA, Université de Nice-Sophia Antipolis, B.P. 93, F-06902 Sophia-Antipolis Cedex, France.

‡Computer Science and Engineering, Arizona State University, PO Box 878809, Tempe, AZ 85287-8809 (charles.colbourn@asu.edu).

§Department of Mathematics, Zhejiang University, Hangzhou 310027, Zhejiang, People's Republic of China.

¶Computer Science, University of Vermont, Burlington, VT 05405.

‖Department of Applied Mathematics, Universidad Politecnica de Catalunya, 08034 Barcelona, Catalonia, Spain.

use the same wavelength (and therefore the same ADM device) in the communication with several nodes. Depending on these choices the total number of ADMs in the network may be reduced. Minimizing the number of ADMs is different from minimizing the number of wavelengths. Indeed, even for the unidirectional ring, the number of wavelengths and the number of ADMs cannot always be simultaneously minimized (see [11, 25] for uniform traffic), although in many cases both parameters can be minimized simultaneously. Both minimization problems have been considered by many authors. See [1, 15] for minimization of the number of wavelengths and [25, 26, 28, 36, 40] for minimization of ADMs. Numerical results, heuristics, and tables have also been given (see, for example, [37]). We consider the particular case of unidirectional rings, so that the routing is unique. There is static uniform symmetric all-to-all traffic, i.e., there is exactly one request of a given size from i to j for each pair (i, j), and no wavelength conversion. With a pair of nodes, $\{i, j\}$, is associated a circle, $C_{\{i,j\}}$, containing both the request from i to j and from j to i. We assume that both requests use the same wavelength. For uniform symmetric traffic in an unidirectional ring, this assumption is not an important restriction and it allows us to focus on the grooming phase independent of the routing. A circle is then a reservation of a fraction of the bandwidth in the whole ring network corresponding to a communication between two nodes. (It is also possible to consider more general classes other than circles containing two symmetric requests packed into the same wavelength. These components are known as circles [11, 40], circuits [37], or primitive rings [13, 14].) If each circle requires only $\frac{1}{C}$ of the bandwidth of a wavelength, we can groom C circles on the same wavelength. C is the *grooming ratio* (or grooming factor). For example, if the request from i to j (and from j to i) is packed in an OC-12 and a wavelength can carry up to an OC-48, the grooming factor is 4. Given the grooming ratio C and the size N of the ring, the objective is to minimize the total number of (SONET) ADMs used, denoted $A(C, N)$. This lowers the network cost by eliminating as many ADMs as possible compared to the no-grooming case.

The problem of minimizing the number of ADMs in a unidirectional ring with uniform traffic can be modeled by graphs, as shown in [5]. Given a unidirectional SONET ring with N nodes, $\overrightarrow{C}_N$, and grooming ratio C, consider the complete graph K_N, i.e., the graph with N vertices in which there is an edge (i, j) for every pair of vertices i and j. The number of edges of K_N equals the number of circles $R = \frac{N(N-1)}{2}$. Moreover, there is a one-to-one mapping between the circles of $\overrightarrow{C}_N$, $C_{\{i,j\}}$ and the edges of K_N, (i, j). Let $\mathcal{S}$ be an assignment of wavelengths and time slots for all requirements among all possible pairs of nodes requiring A ADMs. Let B_ℓ be a subgraph of K_N representing the usage of a given wavelength ℓ in the assignment $\mathcal{S}$. To be precise, let the edges in $E(B_\ell)$ correspond to the circles $C_{\{i,j\}}$ groomed onto the wavelength ℓ, and let the vertices in $V(B_\ell)$ correspond to the nodes of $\overrightarrow{C}_N$ using wavelength ℓ. The number of vertices of B_ℓ, $|V(B_\ell)|$ is the number of nodes using wavelength ℓ or, alternatively, the number of ADMs required for wavelength ℓ. Evidently the total number of edges of B_ℓ, $E(B_\ell)$ is at most the grooming ratio C. With these correspondences the original problem of finding the minimum number of ADMs, $A(C, N)$, required in a ring $\overrightarrow{C}_N$ with grooming ratio C, is equivalent to the following problem in graphs.

PROBLEM 1.1. *Given a number of nodes N and a grooming ratio C, find a partition of the edges of K_N into subgraphs B_ℓ, $\ell = 1, \dots, W$, with $|E(B_\ell)| \leq C$ such that $\sum_{1 \leq \ell \leq W} |V(B_\ell)|$ is minimum.*

In this paper we develop techniques for solving the unidirectional wavelength assignment when the grooming ratio is 6. We determine the exact values of $A(6, N)$

for all values of N except for a finite number of cases.

The paper is organized as follows. In section 2 we introduce some notation and previous results. Section 3 is devoted to the lower bound; in that section we also determine the structure of a decomposition that realizes the lower bound. In section 4, we give constructions that achieve the lower bound for most values of N. That section is divided into three parts. In section 4.1 we show some results from design theory that will be needed later. Section 4.2 is devoted to showing constructions for small cases. Finally, in section 4.3, we give general constructions for all values of N with few exceptions.

2. Previous results.

Optimal constructions for given grooming ratio C have been obtained using tools of graph and design theory [12]. In particular, results are available for grooming ratio $C = 3$ [3], $C = 4$ [6, 28], $C = 5$ [4], and $C \geq N(N-1)/6$ [6]. The problem is also solved for large values of C [6]. Related problems have been studied in both the context of variable traffic requirements [11, 16, 27, 36, 39] and the case of fixed traffic requirements [3, 4, 5, 6, 17, 25, 26, 28, 29, 32, 37, 40].

We now present some results to be used in later sections, leaving specific results on design theory until section 4.1.

Let $\rho(B_\ell)$ denote the ratio for the subgraph B_ℓ, $\rho(B_\ell) = \frac{|E(B_\ell)|}{|V(B_\ell)|}$, and $\rho(m)$ be the maximum ratio of a subgraph with m edges. Let $\rho_{\max}(C)$ denote the maximum ratio of subgraphs with $m \leq C$ edges. We have $\rho_{\max}(C) = \max\{\rho(B_\ell) \mid |E(B_\ell)| \leq C\} = \max_{m \leq C} \rho(m)$. For the sake of illustration, Table 2.1 gives the values of $\rho_{\max}(C)$ for small values of C. For example, for $C = 6$, $\rho_{\max}(6) = \frac{3}{2}$, the bound being attained for K_4.

Theorem 2.1 (see [5]). *Any grooming of R circles with a grooming factor C needs at least $\frac{R}{\rho_{\max}(C)}$ ADMs, i.e., $A(C,N) \geq \frac{N(N-1)}{2\rho_{\max}(C)}$.*

The grooming problem is closely connected to problems in combinatorial design theory. Indeed, an $(N, k, 1)$-design is exactly a partition of the edges of K_N into subgraphs isomorphic to K_k (these are the *blocks* of the design). That corresponds to requiring in our partitioning problem that all the subgraphs B_ℓ be isomorphic to K_k. The classical equivalent definition is, given a set of N elements, find a set of blocks such that each block contains k elements and each pair of elements appears in exactly one block (see [12]). More generally, a G-design of order N (see [12, section IV.22], [7, 8]) consists of a partition of the edges of K_N into subgraphs isomorphic to a given graph G. Our interest in the existence of a G-design is shown by the following proposition.

Proposition 2.2. *If there exists a G-design of order N, where G is a graph with at most C edges and ratio $\rho_{\max}(C)$, then $A(C,N) = \frac{N(N-1)}{2\rho_{\max}(C)}$.*

Necessary conditions 2.3 (existence of a G-design). *If there exists a G-design, then*

(i) $\frac{N(N-1)}{2}$ *is a multiple of $E(G)$,*

(ii) $N-1$ *is a multiple of the greatest common divisor of the degrees of the vertices of G.*

Wilson's theorem [31, 38] establishes that these necessary conditions are also sufficient for large N. From that, given any value of C, for an infinite number of values of N, $A(C,N) = \frac{N(N-1)}{2\rho_{\max}(C)}$. Unfortunately, the values of N for which Wilson's theorem applies are very large. Nevertheless, for small values of C, we can use exact results from design theory. For example, from the existence of G-designs for $G = K_4$ we obtain the following result.

Table 2.1
Values of $\rho_{\max}(C)$ for small C.

C	1	2	3	4	5	6	7	8	9	10
$\rho_{\max}(C)$	$\frac{1}{2}$	$\frac{2}{3}$	1	1	$\frac{5}{4}$	$\frac{3}{2}$	$\frac{3}{2}$	$\frac{8}{5}$	$\frac{9}{5}$	2
C	11	12	13	14	15	16	24	32	48	64
$\rho_{\max}(C)$	2	2	$\frac{13}{6}$	$\frac{14}{6}$	$\frac{5}{2}$	$\frac{5}{2}$	3	$\frac{32}{9}$	$\frac{9}{2}$	$\frac{64}{11}$

Theorem 2.4. *$A(6, N) = \frac{N(N-1)}{3}$ when $N \equiv 1$ or $4 \pmod{12}$.*

The nonexistence of certain G-designs for some values of C and N implies that K_N cannot be optimally decomposed by using isomorphic copies of the same subgraph. This lack of regularity in the decomposition makes it harder to find optimal decompositions and thus to find the value of $A(C, N)$. Furthermore, the solution may be very different for different values of C and N, and Proposition 2.3 suggests that the solutions depend on the congruence class of N.

Theorem 2.1 suggests that the minimum number of ADMs can be achieved by choosing subgraphs such that the average ratio is maximized, or roughly speaking, by choosing subgraphs with a ratio equal to $\rho_{\max}(C)$ whenever possible. Although this last sentence is not to be taken literally, we do show in section 3 that most of the subgraphs in optimal decompositions for $C = 6$ must be isomorphic to K_4.

Even if G-designs do not give a direct solution to our problem, related combinatorial structures assist in the solution. For instance, some types of designs may give a decomposition for a part of the graph or may help constructing solutions by composition from smaller cases.

We introduce specific concepts and results from design theory in section 4.1 in order not to make the presentation overly technical at the outset. See [9, 12] for undefined terms and for a general overview of design theory.

In the remainder of the paper we use standard terms from graph theory. However, let us introduce some notation and terminology that may not be standard. Let $v_1, v_2, \ldots, v_l$ be nonnegative integers; the *complete multipartite graph with class sizes* $v_1, v_2, \ldots, v_l$, denoted $K_{v_1,v_2,\ldots,v_l}$ is the graph with vertex set $V_1 \cup V_2 \cup \cdots \cup V_l$, where $|V_i| = v_i$, and two vertices $x \in V_i$ and $y \in V_j$ are adjacent if and only if $i \neq j$. For $u > 0$, we write $K_{g\times u}$ (resp., $K_{g\times u,m}$) $K_{g,g,\ldots,g}$ (resp., $K_{g,g,\ldots,g,m}$) when g occurs u times.

Given a complete graph K_n, the graph $K_n - e$ is the result of removing one edge. In this paper we also use names for given graphs that are given in Table 3.1.

3. Lower bound for grooming ratio $C = 6$.

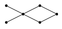
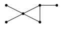
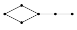
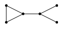
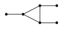
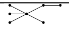
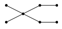
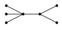
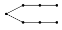
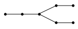
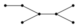
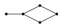
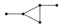
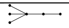
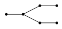
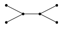
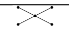
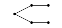
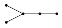
In this section we first give the lower bound for grooming factor $C = 6$ (Theorem 3.1), and then we discuss the possible structure of any decomposition attaining the lower bound.

Theorem 3.1. *Let $R = \frac{N(N-1)}{2}$ denote the number of edges of K_N and A the number of ADMs.*

- *If $N \equiv 1 \pmod 3$, then $A \geq \frac{2R}{3} + \epsilon$, where $\epsilon = 2$ if $N \equiv 7$ or $10 \pmod{12}$ and 0 otherwise.*
- *If $N \equiv 2 \pmod 3$, then $A \geq \frac{2R+N+2}{3}$.*
- *If $N \equiv 0 \pmod 3$, then $A \geq \lceil\frac{6R+2N}{9}\rceil + \epsilon$, where $\epsilon = 1$ if $N \equiv 18, 27 \pmod{36}$, and $\epsilon = 0$ otherwise.*

Table 3.1

Graphs with v vertices and $e \leq 6$ edges. $g_{i,j}$ is the average contribution to degree $\equiv 1 \pmod 3$, $g'_{i,j}$ the average contribution to degree $\equiv 2 \pmod 3$, $\delta_{i,j} = \max_g g_{i,j}$, and $\delta'_{i,j} = \max_{g'} g'_{i,j}$.

Graph		e	v	deg. seq.	$g_{i,j}$		$g'_{i,j}$	
$A_{6,4} = K_4$		6	4	3333	0		0	
$A_{6,5}$		6	5	42222	3	$= \delta_{6,5}$	4.5	$= \delta'_{6,5}$
$B_{6,5}$		6	5	43221	3		3	
$C_{6,5} = K_{3,2}$		6	5	33222	1.5		3	
$D_{6,5}$		6	5	33321	1.5		1.5	
$A_{6,6}$		6	6	522111	4.5	$= \delta_{6,6}$	4.5	$= \delta'_{6,6}$
$B_{6,6}$		6	6	422211	4.5		4.5	
$C_{6,6}$		6	6	432111	4.5		3	
$D_{6,6}$		6	6	322221	3		4.5	
$E_{6,6}$		6	6	332211	3		3	
$F_{6,6}$		6	6	333111	3		1.5	
$A_{6,7}$		6	7	5211111	6	$= \delta_{6,7}$	4.5	
$B_{6,7}$		6	7	4221111	6		4.5	
$C_{6,7}$		6	7	4311111	6		3	
$D_{6,7}$		6	7	6111111	6		3	
$E_{6,7}$		6	7	2222211	4.5		6	$= \delta'_{6,7}$
$F_{6,7}$		6	7	3222111	4.5		4.5	
$H_{6,7}$		6	7	3321111	4.5		3	
$A_{5,4} = K_4 - e$		5	4	3322	1	$= \delta_{5,4}$	2	$= \delta'_{5,4}$
$A_{5,5}$		5	5	42211	4	$= \delta_{5,5}$	3.5	
$B_{5,5}$		5	5	22222	2.5		5	$= \delta'_{5,5}$
$C_{5,5}$		5	5	32221	2.5		3.5	
$D_{5,5}$		5	5	33211	2.5		2	
$A_{5,6}$		5	6	421111	5.5	$= \delta_{5,6}$	3.5	
$B_{5,6}$		5	6	511111	5.5		2.5	
$C_{5,6}$		5	6	322111	4		3.5	
$D_{5,6}$		5	6	331111	4		2	
$E_{5,6}$		5	6	222211	3		5	$= \delta'_{5,6}$
$A_{4,4} = C_4$		4	4	2222	2	$= \delta_{4,4}$	4	$= \delta'_{4,4}$
$B_{4,4}$		4	4	3221	2		2.5	
$A_{4,5}$		4	5	41111	5	$= \delta_{4,5}$	2.5	
$B_{4,5}$		4	5	22211	3.5		4	$= \delta'_{4,5}$
$C_{4,5}$		4	5	32111	3.5		2.5	
$A_{3,3} = C_3$		3	3	222	1.5	$= \delta_{3,3}$	3	$= \delta'_{3,3}$
$A_{3,4}$		3	4	2211	3	$= \delta_{3,4}$	3	$= \delta'_{3,4}$
$B_{3,4}$		3	4	3111	3		1.5	
$A_{2,3}$		2	3	211	2.5	$= \delta_{2,3}$	2	$= \delta'_{2,3}$
$A_{1,2} = K_2$		1	2	11	2	$= \delta_{1,2}$	1	$= \delta'_{1,2}$

Proof. Let $G_{i,j}$ denote a graph with i edges and j vertices. In Table 3.1 are indicated all the possible degree sequences of the connected graphs with $i \leq 6$ (at most six edges) and one example of such a graph. Consider a decomposition of K_N and let $\alpha_{i,j}$ be the number of graphs of type $G_{i,j}$ appearing in the decomposition. We have the two following equations:

$$R = \frac{N(N-1)}{2} = \sum_{i,j} i \cdot \alpha_{i,j}, \tag{3.1}$$

$$A = \sum_{i,j} j \cdot \alpha_{i,j}. \tag{3.2}$$

From (3.1) and (3.2) and the fact that $C = 6$ implies $i \leq 6$, we deduce

$$\begin{aligned} 3A = 2R &+ 3\alpha_{6,5} + 6\alpha_{6,6} + 9\alpha_{6,7} + 2\alpha_{5,4} + 5\alpha_{5,5} + 8\alpha_{5,6} \\ &+ 4\alpha_{4,4} + 7\alpha_{4,5} + 3\alpha_{3,3} + 6\alpha_{3,4} + 5\alpha_{2,3} + 4\alpha_{1,2}. \end{aligned} \tag{3.3}$$

So we always have $A \geq 2R/3$, equality being attained only if there exists a $(N, 4, 1)$-design, which is true only for $N \equiv 1$ or $4 \pmod{12}$ (Theorem 2.4).

Case 1. $N \equiv 1 \pmod 3$.

If $N \equiv 7$ or $10 \pmod{12}$, then $R \equiv 3 \pmod 6$ and the decomposition must contain some graphs having strictly less than six edges. Thus, either it contains at least two subgraphs having less than six edges and then $3A \geq 2R+4$ or only one graph, which is necessarily a C_3; but that is impossible as $K_N - C_3$ cannot be partitioned into K_4, as the three nodes of the C_3 have degree $N-2 \equiv 2 \pmod 3$ (Condition 2.3). Thus we have $A \geq 2R/3 + 2$.

Case 2. $N \equiv 2 \pmod 3$.

The degree of a vertex of K_N is $\equiv 1 \pmod 3$ and so in each vertex we have to use at least either a graph $G_{i,j}$ having a vertex of degree $\equiv 1 \pmod 3$ or two graphs $G_{i,j}$ each having a vertex of degree $\equiv 2 \pmod 3$.

For a graph $G_{i,j}$, let $g^1_{i,j}$ denote its number of vertices of degree $\equiv 1 \pmod 3$ and $g^2_{i,j}$ denote its number of vertices of degree $\equiv 2 \pmod 3$. Write $g_{i,j} = g^1_{i,j} + \frac{1}{2} g^2_{i,j}$. For example, for $A_{6,5}$ (two triangles with a common vertex) the sequence of degrees is 42222 and so $a^1_{6,5} = 1$, $a^2_{6,5} = 4$, and $a_{6,5} = 3$, and for $B_{6,5}$ with degree sequence 43221, $b^1_{6,5} = 2$, $b^2_{6,5} = 2$, and $b_{6,5} = 3$. Values of $g_{i,j}$ are given in Table 3.1.

Now, the condition that the sum of the degrees of a given vertex is $N - 1 \equiv 1 \pmod 3$ implies that

$$\sum_{G_{i,j}} g_{i,j} \geq N. \tag{3.4}$$

Let $\delta_{i,j} = \max_g g_{i,j}$, with the maximum taken over all the graphs with i edges and j vertices. For example, $\delta_{6,5} = 3$ (attained for $A_{6,5}$ and $B_{6,5}$), $\delta_{6,6} = 4.5$ (attained for $A_{6,6}$, $B_{6,6}$, and $C_{6,6}$), and so on.

Equation (3.4) becomes

$$\sum_{i,j} \alpha_{i,j} \delta_{i,j} \geq N. \tag{3.5}$$

That is by using the values of $\delta_{i,j}$

$$\begin{aligned} 3\alpha_{6,5} + 4.5\alpha_{6,6} + 6\alpha_{6,7} + \alpha_{5,4} + 4\alpha_{5,5} + 5.5\alpha_{5,6} + 2\alpha_{4,4} + 5\alpha_{4,5} \\ + 1.5\alpha_{3,3} + 3\alpha_{3,4} + 2.5\alpha_{2,3} + 2\alpha_{1,2} \geq N. \end{aligned} \tag{3.6}$$

Now (3.3) plus inequality (3.6) gives

$$\begin{aligned} 3A \geq 2R + N &+ 1.5\alpha_{6,6} + 3\alpha_{6,7} + \alpha_{5,4} + \alpha_{5,5} + 2.5\alpha_{5,6} \\ &+ 2\alpha_{4,4} + 2\alpha_{4,5} + 1.5\alpha_{3,3} + 3\alpha_{3,4} + 2.5\alpha_{2,3} + 2\alpha_{1,2} \end{aligned} \tag{3.7}$$

and so $A \geq \lceil \frac{2R+N}{3} \rceil$. But, as $N \equiv 2 \pmod 3$ and $R \equiv 1 \pmod 3$, we have $\lceil \frac{2R+N}{3} \rceil = \frac{2R+N+2}{3}$ and finally $A \geq \frac{2R+N+2}{3}$.

Case 3. $N \equiv 0 \pmod 3$.

In this case each vertex of K_N has degree $\equiv 2 \pmod 3$. Thus we have to use in each vertex at least either a graph $G_{i,j}$ having a vertex of degree $\equiv 2 \pmod 3$ or two graphs $G_{i,j}$ each having a vertex of degree $\equiv 1 \pmod 3$.

For a given graph $G_{i,j}$, let us define $g'_{i,j} = g^2_{i,j} + \frac{1}{2}g^1_{i,j}$. For example, for $A_{6,5}$, $a'_{i,j} = 4.5$, but for $B_{6,5}$, $b'_{i,j} = 3$ (values of $g'_{i,j}$ are indicated in Table 3.1).

The condition that the sum of the degrees of a vertex is $N-1 \equiv 2 \pmod 3$ implies that

$$\sum_{G_{i,j}} g'_{i,j} \geq N. \tag{3.8}$$

Let $\delta'_{i,j} = \max_{g'} g'_{i,j}$, with the maximum taken over all graphs with i edges and j vertices. For example, $\delta'_{6,5} = 4.5$ (attained only for $A_{6,5}$).

Equation (3.8) becomes

$$\sum_{i,j} \alpha_{i,j}\delta'_{i,j} \geq N \tag{3.9}$$

or, replacing by the values of $\delta'_{i,j}$,

$$\begin{aligned} &4.5\alpha_{6,5} + 4.5\alpha_{6,6} + 6\alpha_{6,7} + 2\alpha_{5,4} + 5\alpha_{5,5} + 5\alpha_{5,6} + 4\alpha_{4,4} + 4\alpha_{4,5} \\ &\qquad\qquad + 3\alpha_{3,3} + 3\alpha_{3,4} + 2\alpha_{2,3} + \alpha_{1,2} \geq N. \end{aligned} \tag{3.10}$$

Now (3.3) with both sides multiplied by 3 and inequality (3.10) with both sides multiplied by 2 give

$$\begin{aligned} 9A \geq\ &6R + 2N + 9\alpha_{6,6} + 15\alpha_{6,7} + 2\alpha_{5,4} + 5\alpha_{5,5} + 14\alpha_{5,6} + 4\alpha_{4,4} + 13\alpha_{4,5} \\ &+ 3\alpha_{3,3} + 12\alpha_{3,4} + 11\alpha_{2,3} + 10\alpha_{1,2}. \end{aligned} \tag{3.11}$$

As $N \equiv 0 \pmod 3$, we have $6R \equiv 0 \pmod 9$ and we obtain $\lceil \frac{6R+2N}{9} \rceil = \frac{6R+2N+\beta}{9}$, where $\beta = 0$ when $N \equiv 0 \pmod 9$, $\beta = 3$ when $N \equiv 3 \pmod 9$, and $\beta = 6$ when $N \equiv 6 \pmod 9$.

Furthermore, if $N \equiv 3$ or $6 \pmod{12}$, $R \equiv 3 \pmod 6$ and so we cannot use only graphs with six edges. In that case, $9A > 6R + 2N$, in particular if $N \equiv 18$ or $27 \pmod{36}$, we have $A \geq \frac{6R+2N+9}{9}$. □

Let us now examine the possible structure for a decomposition of K_N in order to match the lower bound of Theorem 3.1.

The following remarks are obtained by checking carefully the graphs in Table 3.1 and the equations in the proof of Theorem 3.1.

REMARK 3.2. *When* $N \equiv 7$ *or* $10 \pmod{12}$*, the only way to match the lower bound (*$A = 2R/3 + 2$*) with* $R \equiv 3 \pmod 6$ *and degree* $N - 1 \equiv 0 \pmod 3$ *is by using three subgraphs* $G_{5,4}$ *(that is,* $K_4 - e$*) sharing the two vertices with degree* 3*. It corresponds to a covering of* K_N *by* K_4 *in which an edge is covered four times.*

When $N \equiv 2 \pmod 3$ we distinguish two possible subcases, depending on the congruence class of R. If $N \equiv 2$ or $11 \pmod{12}$, that is, $R \equiv 1 \pmod 6$, the only possibility is $\alpha_{1,2} = 1$ and therefore, we have the next remark.

REMARK 3.3. *Any decomposition attaining the lower bound with* $N \equiv 2$ *or* $11 \pmod{12}$ *must contain one* K_2*,* $\frac{N-2}{3}$ *graphs of type* $A_{6,5}$ *or* $B_{6,5}$*, and the remaining subgraphs being* K_4*.*

If $N \equiv 5$ or $8 \pmod{12}$, that is, $R \equiv 4 \pmod 6$, there are different possibilities.

REMARK 3.4. *Any decomposition attaining the lower bound with* $N \equiv 5$ *or* $8 \pmod{12}$ *must contain* K_4 *and*

- *either* $\alpha_{4,4} = 1$, *that is, one* $A_{4,4}$ *or* $B_{4,4}$ *and* $\frac{N-2}{3}$ $A_{6,5}$ *or* $B_{6,5}$*;*
- *or* $\alpha_{4,5} = 1$, *that is, one* $A_{4,5}$ *(*$B_{4,5}$ *and* $C_{4,5}$ *do not work as* $b_{4,5} = c_{4,5} = 3.5 < 5$*) and* $\frac{N-5}{3}$ $A_{6,5}$ *or* $B_{6,5}$*;*
- *or* $\alpha_{5,4} = 2$, *that is, two* $A_{5,4}$ *(*$K_4 - E$*) and* $\frac{N-2}{3}$ $A_{6,5}$ *or* $B_{6,5}$*;*
- *or* $\alpha_{5,4} = 1$ *and* $\alpha_{5,5} = 1$, *that is, one* $A_{5,4}$, *one* $A_{5,5}$, *and* $\frac{N-5}{3}$ $A_{6,5}$ *or* $B_{6,5}$*;*
- *or* $\alpha_{5,5} = 2$, *that is, two* $A_{5,5}$ *and* $\frac{N-8}{3}$ $A_{6,5}$ *or* $B_{6,5}$.

When $N \equiv 0 \pmod 3$, (3.3), (3.10), and (3.11) can be used to determine the structure of any decomposition attaining the lower bound. Denote by F_4 the graph consisting of two $A_{6,5}$ sharing the same vertex of degree 4 (equivalently, F_4 consists of 4 C_3 having a common vertex). A graph F_4 is decomposed into two $A_{6,5}$ and therefore despite having 9 vertices must be attributed a cost of 10.

The decomposition depends on the congruence class modulo 36 as follows.

REMARK 3.5. *Any decomposition attaining the lower bound must satisfy*

- $N \equiv 0$ *or* $9 \pmod{36}$*: the graph is decomposed into* $\frac{N}{9}$ *vertex disjoint* F_4 *plus* K_4*;*
- $N \equiv 3$ *or* $30 \pmod{36}$*:* $R \equiv 3 \pmod 6$ *implies that* $\alpha_{3,3} = 1$, *and therefore the decomposition contains one* C_3, $\frac{N-3}{9}$ *vertex disjoint* F_4 *plus* K_4.

To obtain the possible decompositions in the remaining cases we use the parameter $g'_{i,j}$ in the inequalities (3.10) and (3.11) to obtain

$$9A \geq \begin{array}{c} 6R + 2N + 3(b'_{6,5} + c'_{6,5}) + 6d'_{6,5} + 2a'_{5,4} \\ + 5b'_{5,5} + 4a'_{4,4} + 3a'_{3,3} + 7\sum g'_{i,j}. \end{array} \tag{3.12}$$

(The last term $\sum g'_{i,j}$ corresponds to graphs $G_{i,j}$ different from K_4, $G_{6,5}$, $A_{5,4}$, $B_{5,5}$, $A_{4,4}$, and $A_{3,3}$.) When $N \equiv 12$ or $21 \pmod{36}$, $R \equiv 0 \pmod 6$ and all subgraphs must contain six edges. Precisely, (3.12) shows the following.

REMARK 3.6. *Any decomposition with* $N \equiv 12$ *or* $21 \pmod{36}$ *that meets the lower bound must contain* K_4 *plus*

- *either a* $C_{6,5}$ *(*$K_{3,2}$*) and* $\frac{N-3}{9}$ F_4 *all vertex disjoint;*
- *or a* $B_{6,5}$ *sharing its vertex of degree* 4 *with an* $A_{6,5}$ *and its vertex of degree* 1 *with another* $A_{6,5}$ *and* $\frac{N-12}{9}$ F_4 *(all these graphs having no other vertices in common);*
- *or five* $A_{6,5}$ *sharing the vertex of degree* 4 *and then* $\frac{N-21}{9}$ F_4*;*
- *or a vertex belonging to four* F_4 *with degree* 2 *in each of them;*
- *or a vertex belonging to three* F_4, *once with degree* 4 *and twice with degree two.*

Similarly, for the remaining cases, we have the next remark.

REMARK 3.7. *Any decomposition with* $N \equiv 6$ *or* $15 \pmod{36}$ *that meets the lower bound must contain* K_4 *plus*

- *either one* C_3 *and same as above (one* $C_{6,5}$ *or* $B_{6,5}$ *or some vertex belonging to five, four, or three* F_4*);*
- *or one* $A_{4,4}$, *one* $A_{5,4}$, *and* $\frac{N-6}{9}$ F_4 *disjoint except for two vertices of degree* 3 *in* $A_{5,4}$*;*
- *or three* $A_{5,4}$ *and* $\frac{N-6}{9}$ F_4, *vertex disjoint except for the six vertices of degree* 3 *in the three* $A_{5,4}$.

REMARK 3.8. *Any decomposition with* $N \equiv 24$ *or* $33 \pmod{36}$ *that meets the lower bound must contain* K_4 *plus*

- *either two* C_3 *and* $\frac{N-6}{9}$ F_4 *vertex disjoint;*
- *or only graphs with six edges like*

- *one* $D_{6,5}$*,*
- *two* $B_{6,5}$ *or* $C_{6,5}$*,*
- *one* $B_{6,5}$ *or* $C_{6,5}$ *with some vertex belonging to five, four, or three* F_4*,*
- *a vertex in eight* $A_{6,5}$ *or two vertices each in four* $A_{6,5}$ *or other combinations with same vertex (or two vertices) belonging to three or more subgraphs.*

REMARK 3.9. *Any decomposition with* $N \equiv 18$ *or* $27 \pmod{36}$ *that meets the lower bound must contain* K_4 *plus*

- *either* $3\ C_3$*,*
- *or one* C_3 *(and some subgraphs as in the preceding case),*
- *or one* $A_{4,4}$ *and one* $B_{5,5}$*,*
- *or* $3\ A_{5,4}$ *and some vertex belonging to three or more subgraphs.*

4. Upper bounds and optimal constructions.

4.1. Some results from design theory.

4.1.1. Definitions and previous results. A *group divisible design* (GDD) is a triple $(X, \mathcal{G}, \mathcal{B})$, where X is a set of points, $\mathcal{G}$ is a partition of X into *groups*, and $\mathcal{B}$ is a collection of subsets of X called *blocks* such that any pair of distinct points from X occur together either in one group or in exactly one block, but not both. A K-GDD of type $g_1^{u_1} g_2^{u_2} \dots g_s^{u_s}$ is a GDD in which every block has size from the set K and in which there are u_i groups of size g_i for $i = 1, 2, \dots, s$.

REMARK 4.1. *The existence of a decomposition of* $K_{g\times u,m}$ *into* K_4 *is equivalent to the existence of a* 4*-GDD of type* $g^u m^1$*.*

A *transversal design* TD(k, g) is a k-GDD of type g^k.

A *pairwise balanced design* (PBD) with parameters $(K; v)$ is a K-GDD of type 1^v. In particular, if $K = k$, a PBD is a G-design with G being the complete graph K_k.

A group divisible design $(X, \mathcal{G}, \mathcal{B})$ is *resolvable* (and referred to as an RGDD) if its block set $\mathcal{B}$ admits a partition into *parallel classes*, each parallel class being a partition of the point set X. A *double group divisible design* (DGDD) is a quadruple $(X, \mathcal{H}, \mathcal{G}, \mathcal{B})$, where X is a set of points, $\mathcal{H}$ and $\mathcal{G}$ are partitions of X (into holes and groups, respectively), and $\mathcal{B}$ is a collection of subsets of X (blocks) such that

(i) for each block $B \in \mathcal{B}$ and each hole $H \in \mathcal{H}$, $|B \cap H| \leq 1$, and

(ii) any pair of distinct points from X which are not in the same hole occur either in some group or in exactly one block, but not both.

A K-DGDD of type $(g_1, h_1^v)^{u_1} (g_2, h_2^v)^{u_2} \dots (g_s, h_s^v)^{u_s}$ is a double group-divisible design in which every block has size from the set K and in which there are u_i groups of size g_i, each of which intersects each of the v holes in h_i points. Thus $g_i = v \cdot h_i$ for $i = 1, 2, \dots, s$. Not every DGDD can be expressed this way, of course, but this is the most general type that we require. One special case, a *modified group divisible design* K-MGDD of type g^u, is a K-DGDD of type $(g, 1^g)^u$. A k-DGDD of type $(g, h^v)^k$ is an incomplete transversal design (ITD) $(k, g; h^v)$ and is equivalent to a set of $k - 2$ holey MOLS of type h^v (see, e.g., [12]).

We recall some known results on designs to be used in subsequent sections.

THEOREM 4.2 (see Theorem 1.27 of [12]). *The multipartite graph* $K_{2\times u}$ *can be partitioned into* $\frac{u(u-1)}{3}$ K_4 *when* $u \equiv 1 \pmod 3$*,* $u > 4$*. Equivalently there exists a* 4*-GDD of type* 2^u*.*

THEOREM 4.3 (see [10] and Chapter 7 of [9]). *The multipartite graph* $K_{g\times 4}$ *can be partitioned into* K_4 *if and only if* $g \neq 2, 6$*. Equivalently there exists a TD*$(4, g)$ *if and only if* $g \neq 2, 6$*.*

The primary recursive construction that we use is Wilson's fundamental construction (WFC) for GDDs (see, e.g., [12]).

Construction 4.4. *Let $(X, \mathcal{G}, \mathcal{B})$ be a GDD, and let $w : X \to \mathbb{Z}^+ \cup \{0\}$ be a weight function on X. Suppose that for each block $B \in \mathcal{B}$, there exists a K-GDD of type $\{w(x) : x \in B\}$. Then there is a K-GDD of type $\{\sum_{x \in G} w(x) : G \in \mathcal{G}\}$.*

We make use of the following existence result.

Theorem 4.5 (see [24]). *There exists a 4-DGDD of type $(mt, m^t)^n$ if and only if $t, n \geq 4$ and $(t-1)(n-1)m \equiv 0 \pmod 3$ except for $(m, n, t) = (1, 4, 6)$ and except possibly for $m = 3$ and $(n, t) \in \{(6, 14), (6, 15), (6, 18), (6, 23)\}$.*

We also make use of the following simple construction for 4-GDDs, which was stated in [23].

Construction 4.6. *If there is a 4-DGDD of type $(g_1, h_1^v)^{u_1}\ (g_2, h_2^v)^{u_2} \ldots (g_s, h_s^v)^{u_s}$, and for each $i = 1, 2, \ldots, s$ there is a 4-GDD of type $h_i^v a^1$, where a is a fixed nonnegative integer, then there is a 4-GDD of type $h^v a^1$, where $h = \sum_{i=1}^{s} u_i h_i$.*

The following results on transversal designs are known (see, for example, [12]).

Theorem 4.7. *A TD(k, g) exists if*

1. *$k = 5$ and $g \geq 4$ and $g \notin \{6, 10\}$;*
2. *$k = 6$ and $g \geq 5$ and $g \notin \{6, 10, 14, 18, 22\}$;*
3. *$k = 7$ and $g \geq 7$ and $g \notin \{10, 14, 15, 18, 20, 22, 26, 30, 34, 38, 46, 60, 62\}$.*

Finally, we make use of the following results on 4-GDDs (see, e.g., [12, 21, 22, 23, 33]).

Theorem 4.8 (see [12, III.1.3, Theorem 1.27]). *Let u and t be positive integers. Then there exists a 4-GDD of type t^u if and only if the conditions in the following table are satisfied:*

Existence of 4-GDDs of Type t^u		
t	u	Necessary and Sufficient Conditions
$1, 5 \pmod 6$	$1, 4 \pmod{12}$	$u \geq 4$
$2, 4 \pmod 6$	$1 \pmod 3$	$u \geq 4,\ (t, u) \neq (2, 4)$
$3 \pmod 6$	$0, 1 \pmod 4$	$u \geq 4$
$0 \pmod 6$	no constraint	$u \geq 4, (t, u) \neq (6, 4)$

Theorem 4.9 (see [12, III.1.3, Theorem 1.28]). *A 4-GDD of type $3^u m^1$ exists if and only if either $u \equiv 0 \mod 4$ and $m \equiv 0 \mod 3$, $0 \leq m \leq (3u-6)/2$; or $u \equiv 1 \mod 4$ and $m \equiv 0 \mod 6$, $0 \leq m \leq (3u-3)/2$; or $u \equiv 3 \mod 4$ and $m \equiv 3 \mod 6$, $0 < m \leq (3u-3)/2$.*

Theorem 4.10 (see [21, Theorem 1.7]). *There exists a 4-GDD of type $g^4 m^1$ with $m > 0$ if and only if $g \equiv m \equiv 0 \mod 3$ and $0 < m \leq \frac{3g}{2}$.*

Theorem 4.11 (see [22, Theorem 1.6]). *There exists a 4-GDD of type $6^u m^1$ for every $u \geq 4$ and $m \equiv 0 \mod 3$ with $0 \leq m \leq 3u - 3$ except for $(u, m) = (4, 0)$ and except possibly for $(u, m) \in \{(7, 15), (11, 21), (11, 24), (11, 27), (13, 27), (13, 33), (17, 39), (17, 42), (19, 45), (19, 48), (19, 51), (23, 60), (23, 63)\}$.*

Theorem 4.12 (see [18, Theorem 3.16]). *There exists a 4-GDD of type $12^u m^1$ for each $u \geq 4$ and $m \equiv 0 \mod 3$ with $0 \leq m \leq 6(u-1)$.*

We also employ current existence results on 4-RGDDs.

Theorem 4.13 (see [19, 20]). *The necessary conditions for the existence of a 4-RGDD(t^u), namely, $u \geq 4$, $tu \equiv 0 \pmod 4$ and $t(u-1) \equiv 0 \pmod 3$, are also sufficient except for $(t, u) \in \{(2, 4), (2, 10), (3, 4), (6, 4)\}$ and possibly excepting*

1. $t \equiv 2, 10 \pmod{12}$*:* $t = 2$ *and* $u \in \{34, 46, 52, 70, 82, 94, 100, 118, 130, 142, 178, 184, 202, 214, 238, 250, 334, 346\}$*;* $t = 10$ *and* $u \in \{4, 34, 52, 94\}$*;* $t \in [14, 454] \cup \{478, 502, 514, 526, 614, 626, 686\}$ *and* $u \in \{10, 70, 82\}$*;*
2. $t \equiv 6 \pmod{12}$*:* $t = 6$ *and* $u \in \{6, 54, 68\}$*;* $t = 18$ *and* $u \in \{18, 38, 62\}$*;*
3. $t \equiv 9 \pmod{12}$*:* $t = 9$ *and* $u = 44$*;*
4. $t \equiv 0 \pmod{12}$*:* $t = 12$ *and* $u = 27$*;* $t = 36$ *and* $u \in \{11, 14, 15, 18, 23\}$.

4.1.2. Existence of 4-GDDs of type $36^u m^1$, for small values of m. Here we consider 4-GDDs of type $36^u m^1$ with $m \in \{3, 6, 9, \ldots, 33\}$. Whenever we refer to a 4-RGDD of type g^u, the existence of such RGDDs comes from Theorem 4.13.

LEMMA 4.14. *There exists a* 4*-GDD of type* $36^u m^1$ *for each* $u \geq 4$*,* $u \equiv 0, 1, 3 \bmod 4$ *and* $m \in \{3, 6, 9, \ldots, 33\}$.

Proof. Start with a TD$(5, u)$ and adjoin an infinite point ∞ to the groups, then delete a finite point so as to form a $\{5, u+1\}$-GDD of type $4^u u^1$. Each block of size $u + 1$ intersects the group of size u in the infinite point ∞ and each block of size 5 intersects the group of size u, but certainly not in ∞. Now, in the group of size u, we give ∞ weight 0 (when $u \equiv 0, 1 \bmod 4$) or 3 (when $u \equiv 3 \bmod 4$) and give the remaining points weight 0, 3, 6, 9, or 12. Give all other points in the $\{5, u+1\}$-GDD weight 9. Replace the blocks in the $\{5, u+1\}$-GDD by 4-GDDs of types 9^u, $9^u 3^1$, or $9^4(3i)^1$ (from Theorem 4.10) with $i \in \{0, 1, 2, 3, 4\}$ to obtain the 4-GDDs. Here, the input designs that are 4-GDDs of type $9^u 3^1$ when $u \equiv 3 \bmod 4$ come from [23]. □

This leaves only the case for $u \equiv 2 \bmod 4$ to consider.

LEMMA 4.15. *There exists a* 4*-GDD of type* $36^6 m^1$ *for each* $m \in \{3, 6, 9, \ldots, 33\}$.

Proof. For $m \in \{3, 6, 9, 12, 15\}$, starting from a 4-DGDD of type $(36, 6^6)^6$ from Theorem 4.5 and applying Construction 4.6 with 4-GDDs of type $6^6 m^1$ to fill in holes, we obtain the designs. For other values of m, start from a TD$(7, 9)$ and apply WFC with weight 4 to the points in the first six groups and weight 1 or 4 to the remaining points. The 4-GDD of type $4^6 1^1$ is from [30, 23]. □

LEMMA 4.16. *There exists a* 4*-GDD of type* $36^{10} m^1$ *for each* $m \in \{3, 6, 9, \ldots, 144\}$.

Proof. Complete the 12 parallel classes of a 4-RGDD of type 4^{10} to obtain a 5-GDD of type $4^{10} 12^1$. Apply WFC and give weight 9 to the points in the groups of size 4 and weight 0, 3, 6, 9, or 12 to the remaining points. The result follows from Theorem 4.10. □

LEMMA 4.17. *There exists a* 4*-GDD of type* $36^{14} m^1$ *for each* $m \in \{3, 6, 9, \ldots, 48\}$.

Proof. Take a 5-GDD of 4^{15} and apply WFC with weight 9 to the points in the first 14 groups and weight 0, 3, 6, 9, or 12 to the remaining points. □

LEMMA 4.18. *There exists a* 4*-GDD of type* $36^{18} m^1$ *for each* $m \in \{3, 6, 9, \ldots, 48\}$.

Proof. Take a $(77, \{5, 9^*\}, 1)$-PBD (the existence of such a PBD follows from [2]) and remove a point not in the single block of size 9 to obtain a $\{5, 9\}$-GDD of type 4^{19}. The single block of size 9 can hit only 9 groups of the GDD. Apply WFC with weight 9 to the points in the first 18 groups such that the single block of size 9 is covered by them and weight 0, 3, 6, 9, or 12 to the remaining points. □

LEMMA 4.19. *There exists a* 4*-GDD of type* $36^{22} m^1$ *for each* $m \in \{3, 6, 9, \ldots, 336\}$.

Proof. Complete the 28 parallel classes of a 4-RGDD of type 4^{22} to obtain a 5-GDD of type $4^{22} 28^1$. Apply WFC and give weight 9 to the points in the groups of size 4 and weight 0, 3, 6, 9, or 12 to the remaining points. □

LEMMA 4.20. *There exists a* 4*-GDD of type* $36^u m^1$ *for each* $u \geq 26$ *and* $u \equiv 2 \bmod 4$ *with* $m \in \{3, 6, 9, \ldots, 33\}$.

Proof. Suppose that $u = 4s + 2$ and $s \geq 6$. Take a 4-GDD of type $(36s - 36)^4(216 + m)^1$ from Theorem 4.10 and fill in 4-GDDs of type $36^{(s-1)}$ and 4-GDDs of type $36^6 m^1$ to obtain the 4-GDDs. □

Combining Lemmas 4.14–4.20, we have the following.

THEOREM 4.21. *There exists a 4-GDD of type $36^u m^1$ for each $u \geq 4$ with $m \in \{3, 6, 9, \dots, 33\}$.*

4.1.3. Existence of 4-GDDs of type $36^u m^1$, for large values of m and other types.

Now we consider 4-GDDs of type $36^u m^1$ with $m \in \{117, 822, 840, 846, 852\}$.

LEMMA 4.22. *There exists a 4-GDD of type $36^u m^1$ for each $u \geq 7$,*

$$u \notin U = \{10, 14, 15, 18, 20, 22, 26, 30, 34, 38, 46, 60, 62\}$$

and $m \equiv 0 \mod 3$ with $0 \leq m \leq 18u - 18$.

Proof. Start with a TD$(7, u)$ and adjoin an infinite point ∞ to the groups, then delete a finite point so as to form a $\{7, u+1\}$-GDD of type $6^u u^1$. Each block of size $u+1$ intersects the group of size u in the infinite point ∞ and each block of size 7 intersects the group of size u, but certainly not in ∞. Now, in the group of size u, we give ∞ weight 0 or $3u-3$ and give the remaining points weight 0, 3, 6, 9, 12, or 15. Give all other points in the $\{7, u+1\}$-GDD weight 6. Replace the blocks in the $\{7, u+1\}$-GDD by 4-GDDs of types 6^u, $6^u(3u-3)^1$ or $6^6(3i)^1$ with $i \in \{0, 1, 2, 3, 4, 5\}$ to obtain the 4-GDDs. Here, the input 4-GDDs all come from Theorem 4.11. □

Recall that a necessary condition for the existence of a 4-GDD of type $g^u m^1$ is that $u >= 2m/g + 1 > 0$ (see [12]). This leaves the cases for $m = 117$ and $u \in U$ as well as $m = 822, 840, 846, 852$ and $u = 60, 62$ to treat.

LEMMA 4.23. *There exists a 4-GDD of type $36^u 117^1$ for each $u \in U$.*

Proof. For $u = 10$, the proof follows from Lemma 4.16. For other values of u, start from a 4-RGDD of type 12^u and complete all the parallel classes to obtain a 5-GDD of type $12^u(4u-4)^1$. Give weight 0 or 3 to the points in the group of size $4u-4$ and weight 3 to the remaining points. □

LEMMA 4.24. *There exists a 4-GDD of type $36^u m^1$ for each $u = 60, 62$ and $m = 822, 840, 846$.*

Proof. Take a 4-GDD of type $6^{\frac{u}{2}} 69^1$ from Theorem 4.11 and adjoin an infinite point ∞ to the groups, then delete a finite point in the group of size 69 so as to form a $\{4, 7\}$-GDD of type $3^u 69^1$. Each block of size 7 intersects the group of size 69 in the infinite point ∞, while each block of size 4 does not. Now, we give ∞ weight $0, 3, \dots, 27$ or 30 and give all the remaining points weight 12. Replace the blocks in the $\{4, 7\}$-GDD by 4-GDDs of types 12^4 or $12^6 i^1$ with $i \in \{0, 3, 6, \dots, 30\}$ from Theorem 4.12 to obtain the 4-GDDs. □

We still have $m = 852$ and $u \in \{60, 62\}$ to handle.

LEMMA 4.25. *There exists a 4-GDD of type $36^u 852^1$ for each $u \in \{60, 62\}$.*

Proof. For $u = 60$, the proof is similar to that of Lemma 4.23. Here, we employ a 4-RGDD of type 6^{60}. For $u = 62$, take a resolvable 3-RGDD of type 12^{62} and apply weight 3, using resolvable 3-MGDDs of type 3^3 to obtain a resolvable 3-DGDD of type $(36, 12^3)^{62}$. Adjoin 732 infinite points to complete the parallel classes and then adjoin a further 120 ideal points, filling in 4-GDDs of type $12^{62} 120^1$ from Theorem 4.12, to obtain a 4-GDD of type $36^{62}(732 + 120)^1$. □

Combining Lemmas 4.22–4.25, together with the fact that a necessary condition for the existence of a 4-GDD of type $g^u m^1$ is that $u >= 2m/g + 1 > 0$ (see [12]), we obtain the following result.

THEOREM 4.26.

1. *There exists a 4-GDD of type $36^u 117^1$ if and only if $u \geq 8$.*
2. *There exists a 4-GDD of type $36^u 822^1$ if and only if $u \geq 47$.*

3. *There exists a* 4*-GDD of type* $36^u m^1$ *with* $m = 840, 846$ *if and only if* $u \geq 48$.
4. *There exists a* 4*-GDD of type* $36^u 852^1$ *if and only if* $u \geq 49$.

Here we collect some partial results with $g = 117$ to be used later.

Lemma 4.27. *There exists a* 4*-GDD of type* $117^7 m^1$ *for* $m \in \{3, 21, 27, 33\}$.

Proof. A 4-GDD of type $117^7 3^1$ appears in [23]. A 4-GDD of type $9^7 27^1$ appears in [23]. So fill one set of groups in a 4-DGDD of type $(117, 9^{13})^7$ from [24] to obtain a 4-GDD of type $117^7 27^1$.

For $117^7 33^1$, start from a 4-GDD of type $12^7 33^1$ and give weight 7 to each point, using 4-MGDDs of type 7^4. This gives a 4-DGDD of type $(84, 12^7)^7 (231, 33^7)^1$. Adjoining 33 infinite points and filling in 4-GDDs of type $12^7 33^1$ and a 4-GDD of type 33^8, we obtain a 4-GDD of type $117^7 33^1$. Similarly, we can start from a 4-GDD of type $12^8 21^1$ to obtain a 4-GDD of type $117^7 21^1$. □

4.2. Optimal constructions for small cases. We include in this section constructions for small cases to be used in the general theorems. In this discussion, we denote the graph $A_{6,5}$ as $\{A, B, C, D, E\}$, where A is the vertex of degree 4 and where $\{B, C\}$ and $\{D, E\}$ are edges; we denote the graph $B_{6,5}$ as $\{A, B, C, D, E\}$, where A is the vertex of degree 4, C is the vertex of degree 3, B and D the vertices of degree 2 (joined to A and C), and E is the vertex of degree 1.

Let us start this section with a trivial result.

Lemma 4.28. *The lower bound is attained for* $N \leq 6$*, i.e.,* $A(6, 2) = 2$*,* $A(6, 3) = 3$*,* $A(6, 4) = 4$*,* $A(6, 5) = 9$*, and* $A(6, 6) = 12$.

Let us recall that the lower bound also holds for $N \equiv 1$ or 4 (mod 12) by Theorem 2.4.

We have the following results for small values of N.

Lemma 4.29. *The lower bound is not attained for* $N = 7$*. Moreover,* $A(6, 7) = 17$.

Proof. The partition is obtained using the two K_4 $\{0, 1, 2, 3\}$ and $\{0, 4, 5, 6\}$, the $K_{2,3}$ between nodes $1, 2$ and $4, 5, 6$, and the $K_{1,3}$ between node 3 and nodes $4, 5, 6$.

An exhaustive search establishes that no decomposition exists with cost 16. □

Lemma 4.30. *The lower bound is realized for* $N = 8$*, i.e.,* $A(6, 8) = 22$.

Proof. Let the vertices of K_8 be $V_8 = \{i,\ i \in \mathbb{Z}_8\}$. The decomposition consists of two K_4 $\{0, 1, 2, 3\}$ and $\{0, 4, 5, 6\}$, two $B_{6,5}$ $\{\{1, 4\}, \{1, 5\}, \{1, 6\}, \{1, 7\}, \{4, 7\}, \{5, 7\}\}$ and $\{\{0, 7\}, \{2, 7\}, \{3, 7\}, \{6, 7\}, \{2, 6\}, \{3, 6\}\}$, and the C_4 $(2, 4, 3, 5)$. □

Lemma 4.31. *The lower bound is not attained for* $N = 9$*. Moreover,* $A(6, 9) = 27$.

Proof. The general lower bound gives $A(6, 9) \geq 26$. However, to obtain $A(6, 9) = 26$, K_9 can be partitioned into one F_4 and four K_4, but $K_9 - F_4$ is $K_{2,2,2,2}$, which cannot be decomposed into K_4. Thus $A(6, 9) \geq 27$.

Furthermore, a partition of K_9 is obtained using the three K_4 with vertex sets

$$\{0, 4, 5, 6\}, \{0, 3, 7, 8\}, \{1, 2, 3, 6\},$$

plus the three $K_{2,3}$ $\{3i + 1, 3i + 2 | 3i, 3(i + 1) + 1, 3(i + 1) + 2\}$, $i = 0, 1, 2$, indices taken modulo 9. So altogether $A(6, 9) = 27$. □

Lemma 4.32. *The lower bound is not attained for* $N = 10$*. Moreover,* $A(6, 10) = 34$.

Proof. First we establish that $A(6, 10) \leq 34$. Form three K_4 meeting in the element 9.

The remaining edges form $K_{3,3,3}$ on vertex set $\{0, \ldots, 8\}$. Suppose that $\{0, 1, 2\}$ is one class of the tripartition. Choose a matching $\{a_1, b_1\}, \{a_2, b_2\}, \{a_3, b_3\}$ on the vertices $\{3, \ldots, 8\}$ and for $i = 1, 2, 3$, form a $K_4 - e$ on $\{0, 1, a_i, b_i\}$ omitting the edge $\{0,1\}$. The remaining 12 edges form a 6-wheel (a 6-cycle with a seventh vertex attached to each of the six). This can be decomposed into two copies of $D_{6,5}$.

There are three 4-vertex 6-edge graphs, three 4-vertex 5-edge graphs, and two 5-vertex 6-edge graphs in this partition, for a total of 34.

Any solution of cost less than 34 must have at least four K_4 by (3.3), and there is a unique way up to isomorphism to place four K_4. An exhaustive examination establishes that no such decomposition has cost less than 34. □

Lemma 4.33. *The lower bound is realized for $N = 11$, i.e., $A(6,11) = 41$.*

Proof. Let the vertices of K_{11} be $V_{11} = \{\alpha\} \cup \{\beta\} \cup \{x_i^j,\ i,j \in \mathbb{Z}_3\}$. The decomposition consists of the K_2 $\{\alpha,\beta\}$, plus the three $A_{6,5}$ $\left\{x_i^0, x_{i+1}^1, x_{i+2}^2, x_{i+1}^2, x_{i+2}^1\right\}$, $i = 0,1,2$, plus the three K_4 $\left\{\alpha, x_i^0, x_i^1, x_i^2\right\}$, $i = 0,1,2$, plus the three K_4 $\left\{\beta, x_0^j, x_1^j, x_2^j\right\}$, $j = 0,1,2$. □

Lemma 4.34. *The lower bound is not attained for $N = 12$. Moreover, $A(6,12) = 48$.*

Proof. The general lower bound gives $A(6,12) \geq 47$. However, to obtain $A(6,12) = 47$, there must be 11 6-vertex graphs in the decomposition. The only way in which nine of these can be K_4 leaves four K_3, so we need only consider situations with eight K_4 and three 6-edge graphs on five vertices. An exhaustive search establishes that no such decomposition exists. Thus $A(6,12) \geq 48$.

Let $V = \sum_{i=1}^{4} V_i$ with $|V_i| = 3$; then $K_{3\times 4}$ can be partitioned into nine K_4 (Theorem 4.3). Thus a partition of K_{12} uses nine K_4 and four C_3. So altogether $A(6,12) = 48$. □

Lemma 4.35. *The lower bound is realized for $N = 14$, i.e., $A(6,14) = 66$.*

Proof. Let the vertices of K_{14} be $V_{14} = \{\alpha\} \cup \{\beta\} \cup \{x_i^j,\ i \in \mathbb{Z}_4, j \in \mathbb{Z}_3\}$. The decomposition consists of the K_2 $\{\alpha,\beta\}$, plus the four $B_{6,5}$ $\left\{x_i^0, x_{i+2}^1, x_{i+1}^2, x_{i+3}^1, x_{i+3}^2\right\}$, $i = 0,1,2,3$, plus the 11 K_4 $\{x_0^j, x_1^j, x_2^j, x_3^j\}$, $j = 0,1,2$, $\left\{\alpha, x_i^0, x_i^1, x_i^2\right\}$, $i = 0,1,2,3$, $\left\{\beta, x_i^0, x_{i+1}^1, x_{i+2}^2\right\}$, $i = 0,1,2,3$. □

The next lemma enables us to determine that the lower bound is attained for several values of N.

Lemma 4.36. *When $N = 2t+1$, $t \equiv 1 \pmod 3$, $t > 4$, then $A(6,N) \leq 4\frac{t(t-1)}{3} + 5\left\lfloor \frac{t}{2} \right\rfloor + \epsilon$, where $\epsilon = 3$ if t is odd and 0 otherwise.*

Proof. Let the vertices be α and x_i^j for $i \in \mathbb{Z}_2$ and $j \in \mathbb{Z}_t$. A partition of K_{2t+1} consists of a partition of $K_{2\times t}$ into $\frac{t(t-1)}{3}$ K_4 (Theorem 4.2), plus $\left\lfloor \frac{t}{2} \right\rfloor$ G_5, each one formed as the union of the two C_3 $\left\{\alpha, x_0^{2k}, x_1^{2k}\right\}$ and $\left\{\alpha, x_0^{2k+1}, x_1^{2k+1}\right\}$, $k = 0,1,\ldots,\left\lfloor \frac{t}{2} \right\rfloor/2-1$, and plus the C_3 $\left\{\alpha, x_0^{t-1}, x_1^{t-1}\right\}$ when t is odd. So altogether we have $A(6,2t+1) \leq 4\frac{t(t-1)}{3} + 5\left\lfloor \frac{t}{2} \right\rfloor + \epsilon$, where $\epsilon = 3$ if t is odd and 0 otherwise. □

Corollary 4.37. *The lower bound is realized for $N \in \{15,21,27,33\}$: $A(6,15) = 74$, $A(6,21) = 145$, $A(6,27) = 241$ and $A(6,33) = 360$.*

Proof. Application of Lemma 4.36 and Theorem 3.1. □

Lemma 4.38. *The lower bound is not attained for $N = 19$. Moreover, $A(6,19) = 119$.*

Proof. We first establish that $A(6,19) \leq 119$. Partition K_{19} into 25 K_4,

$$\begin{gathered}\{\{0,1,2,4\}, \{0,3,5,6\}, \{0,7,8,9\}, \{0,10,11,12\}, \{0,13,14,15\},\\ \{0,16,17,18\}, \{1,3,7,10\}, \{1,5,8,11\}, \{1,6,13,16\}, \{1,9,14,17\},\\ \{1,12,15,18\}, \{2,3,8,15\}, \{2,5,9,18\}, \{2,6,10,17\}, \{2,7,12,13\},\\ \{2,11,14,16\}, \{3,4,14,18\}, \{3,9,12,16\}, \{3,11,13,17\}, \{4,5,12,17\},\\ \{4,6,9,15\}, \{5,10,15,16\}, \{6,7,11,18\}, \{6,8,12,14\}, \{8,10,13,18\}\},\end{gathered}$$

and four other graphs,

$D_{6,5}$: {{4,7},{4,8},{4,16},{7,16},{8,16},{8,17}},
$C_{6,5}$: {{4,10},{4,11},{4,13},{9,10},{9,11},{9,13}},
$D_{5,5}$: {{5,7},{5,13},{5,14},{7,14},{10,14}}, and
C_4: {{7,15},{7,17},{11,15},{15,17}}.

A maximum packing of K_4 in K_{19} has 25 K_4, but the example in [35] does not leave edges having a partition with cost 19, as this example does. Indeed, exhaustive computation showed that there are 249 nonisomorphic graphs that can be left by taking 25 K_4 from K_{19}. None yields a graph with cost less than 19. The only remaining possibility is to choose 24 K_4, three 5-edge 4-vertex graphs, and two 6-edge 5-vertex graphs, but a further exhaustive computation yielded no such partition. □

LEMMA 4.39. *The lower bound is realized for $N = 20$, i.e., $A(6, 20) = 134$.*

Proof. Let the vertices of K_{20} be $V = V_1 \cup V_2$ with $|V_1| = 5$ and $|V_2| = 15$, and let the vertices of V_1 be $\{i,\ i \in \mathbb{Z}_5\}$.

The K_{15} on V_2 can be partitioned into seven parallel classes $\mathcal{C}_j$, $j \in \mathbb{Z}_7$, each consisting of five triangles $\mathcal{C}_{j,k}$, $k \in \mathbb{Z}_5$, by the existence of a resolvable $(15, 3, 1)$-design.

For $i \in \mathbb{Z}_5$, we construct five K_4 built on node i and class $\mathcal{C}_{i,k}$, so altogether we have 25 K_4. Furthermore, the 10 triangles of the classes $\mathcal{C}_5$ and $\mathcal{C}_6$ can be joined in pairs to form five graphs isomorphic to $A_{6,5}$ (since there exist five vertices each belonging to exactly one triangle of $\mathcal{C}_5$ and one of $\mathcal{C}_6$). Finally, the K_5 on V_1 can be decomposed into one C_4 and one $A_{6,5}$. Altogether we have decomposed K_{20} into 1 C_4, 6 $A_{6,5}$, and 25 K_4. □

LEMMA 4.40. *The lower bound is realized for $N = 23$, i.e., $A(6, 23) = 177$.*

Proof. Let the vertices of K_{23} be $\{\alpha\} \cup \{\beta\} \cup \{x_i^j,\ i \in \mathbb{Z}_7, j \in \mathbb{Z}_3\}$. The decomposition consists of the K_2 $\{\alpha, \beta\}$, plus the 7 $A_{6,5}$ $\left\{x_i^0, x_i^1, x_i^2, x_{i+1}^1, x_{i+2}^2\right\}$, $i \in \mathbb{Z}_7$, and the 35 K_4,

$$\left\{\alpha, x_i^0, x_{i+2}^1, x_{i+4}^2\right\}, \left\{\beta, x_i^0, x_{i+4}^1, x_{i+1}^2\right\}, \left\{x_i^0, x_{i+3}^1, x_{i+5}^1, x_{i+6}^1\right\},$$

$$\left\{x_i^1, x_{i+3}^2, x_{i+5}^2, x_{i+6}^2\right\}, \text{ and } \left\{x_i^2, x_{i+1}^0, x_{i+2}^0, x_{i+4}^0\right\} \text{ for } i \in \mathbb{Z}_7. \qquad \square$$

LEMMA 4.41. *The lower bound is realized for $N = 26$, i.e., $A(6, 26) = 226$.*

Proof. Let the vertices of K_{26} be $\{\alpha\} \cup \{\beta\} \cup \{x_i^j,\ i \in \mathbb{Z}_8, j \in \mathbb{Z}_3\}$. The decomposition consists of the K_2 $\{\alpha, \beta\}$, plus the 8 $A_{6,5}$ $\left\{x_i^0, x_{i+5}^2, x_{i+6}^1, x_{i+2}^2, x_{i+7}^1\right\}$, plus the 16 K_4 $\left\{\alpha, x_i^0, x_i^1, x_i^2\right\}$ and $\left\{\beta, x_i^0, x_{i+1}^2, x_{i+3}^1\right\}$, $i \in \mathbb{Z}_8$, plus the 24 K_4 $\{x_i^j, x_{i+1}^j, x_{i+2}^{j+1}, x_{i+5}^{j+1}\}$, $i \in \mathbb{Z}_8$ and $j \in \mathbb{Z}_3$, and plus the 6 K_4 $\{x_i^j, x_{i+2}^j, x_{i+4}^j, x_{i+6}^j\}$, $i = 0, 1$ and $j \in \mathbb{Z}_3$. □

LEMMA 4.42. *The lower bound is realized for $N = 29$, i.e., $A(6, 29) = 281$.*

Proof. Let $V = V_1 \cup V_2$ with $|V_1| = 8$ and $|V_2| = 21$, and let the vertices of V_1 be $\{i,\ i \in Z_8\}$.

The K_8 on V_1 can be decomposed into one C_4, 2 $B_{6,5}$, and 2 K_4. The K_{21} on V_2 can be partitioned into 10 parallel classes $\mathcal{C}_j$, $j \in \mathbb{Z}_{10}$, each consisting of 7 triangles $\mathcal{C}_{j,k}$, $k \in \mathbb{Z}_7$, by the existence of a resolvable $(21, 3, 1)$-design. Finally, like for $N = 20$ (Lemma 4.39), we build for each $i \in \mathbb{Z}_8$, 7 K_4 on node i and class $\mathcal{C}_{i,k}$, so altogether 56 K_4; then we pair two by two the triangles of the last two classes $\mathcal{C}_8$ and $\mathcal{C}_9$ to obtain 7 $A_{6,5}$. Altogether we have decomposed K_{29} into 1 C_4, 9 graphs of type $A_{6,5}$ or $B_{6,5}$, and 58 K_4. □

LEMMA 4.43. *The lower bound is realized for $N = 32$, i.e., $A(6, 32) = 342$.*

Proof. Let the vertices of K_{32} be $\{\alpha, \beta, \gamma, \delta, \epsilon\} \cup V_1 \cup V_2 \cup V_3$, where $|V_j| = 9$, $J = 0, 1, 2$, and $V_j = \{x_i^j,\ i \in \mathbb{Z}_9\}$. The K_9 on V_j can be partitioned into four parallel classes $\mathcal{C}_k^j$, $k \in \mathbb{Z}_4$, each consisting of three triangles $\mathcal{C}_{k,l}^j$, $k \in \mathbb{Z}_4$, by the existence of a resolvable $(9, 3, 1)$-design. Let $\mathcal{C}_3^j = \{\{x_i^j, x_{3+i}^j, x_{6+i}^j\},\ i \in \mathbb{Z}_3\}$.

As for $N = 20$ (Lemma 4.39), we build for α 9 K_4 with classes $\mathcal{C}_0^j$, $j = 0, 1, 2$, for β 9 K_4 with classes $\mathcal{C}_1^j$, $j = 0, 1, 2$, and for γ 9 K_4 with classes $\mathcal{C}_2^j$, $j = 0, 1, 2$, so altogether 27 K_4. We also build the 45 K_4 $\left\{x_i^0, x_{i+3}^0, x_{i+4}^1, x_{i+4}^2\right\}$, $\left\{x_i^0, x_{i+2}^1, x_{i+8}^1, x_i^2\right\}$, $\left\{x_i^0, x_i^1, x_{i+2}^2, x_{i+5}^2\right\}$, $\left\{\delta, x_i^0, x_{i+3}^1, x_{i+7}^2\right\}$, and $\left\{\epsilon, x_i^0, x_{i+5}^1, x_{i+8}^2\right\}$, $i \in \mathbb{Z}_9$, and the 9 $A_{6,5}$ $\left\{x_i^0, x_{i+6}^1, x_{i+3}^2, x_{i+7}^1, x_{i+6}^2\right\}$, $i \in \mathbb{Z}_9$. Finally the K_5 on $\{\alpha, \beta, \gamma, \delta, \epsilon\}$ can be decomposed into a C_4 and one $A_{6,5}$. Altogether we have decomposed K_{32} into 1 C_4, 10 $A_{6,5}$, and 72 K_4. □

Lemma 4.44. *The lower bound is realized for* $N = 35$, *i.e.,* $A(6, 35) = 409$.

Proof. Let the vertices of K_{35} be $\{\alpha\} \cup \{\beta\} \cup \{x_i^j,\ i \in \mathbb{Z}_{11}, j \in \mathbb{Z}_3\}$. The decomposition consists of the K_2 $\{\alpha, \beta\}$, plus the 11 $A_{6,5}$ $\left\{x_i^0, x_{i+3}^1, x_{i+5}^2, x_{i+6}^1, x_{i+6}^2\right\}$, $i \in \mathbb{Z}_{11}$, plus the 88 K_4,

$$\left\{\alpha, x_i^0, x_{i+1}^1, x_{i+2}^2\right\}, \left\{\beta, x_i^0, x_{i+2}^1, x_{i+7}^2\right\}, \left\{x_i^0, x_{i+7}^1, x_{i+8}^1, x_{i+10}^1\right\},$$

$$\left\{x_i^1, x_{i+6}^2, x_{i+7}^2, x_{i+9}^2\right\}, \left\{x_i^2, x_i^0, x_{i+2}^0, x_{i+10}^0\right\}, \left\{x_i^0, x_{i+4}^0, x_{i+4}^1, x_{i+9}^1\right\},$$

$$\left\{x_i^1, x_{i+4}^1, x_{i+3}^2, x_{i+8}^2\right\}, \left\{x_i^0, x_{i+5}^0, x_{i+4}^2, x_{i+8}^2\right\}$$

for $i \in \mathbb{Z}_{11}$. □

Lemma 4.45. *The lower bound is realized for* $N = 36$, *i.e.,* $A(6, 36) = 428$.

Proof. First recall that K_{12} can be partitioned into four disjoint C_3 plus nine K_4. Thus let the vertices of K_{12} be labeled α_i, $i \in \mathbb{Z}_4$, and x_j, $j \in \mathbb{Z}_8$, such that $\{\alpha_0, \alpha_1, \alpha_2, \alpha_3\}$ is one K_4 and the four C_3 are $\{\alpha_i, x_{2i}, x_{2i+1}\}$, $i \in \mathbb{Z}_4$.

Now let the 36 vertices be α_i, $i \in \mathbb{Z}_4$, and x_j^k, $j \in \mathbb{Z}_8$, and $k \in \mathbb{Z}_4$, and let $V_k = \left\{x_j^k,\ j \in \mathbb{Z}_8\right\}$.

A partition of K_{36} uses

- the K_4 $\{\alpha_0, \alpha_1, \alpha_2, \alpha_3\}$;
- eight $A_{6,5}$, each the union of two C_3 $\left\{\alpha_i, x_{2i}^{2k}, x_{2i+1}^{2k}\right\}$ and $\left\{\alpha_i, x_{2i}^{2k+1}, x_{2i+1}^{2k+1}\right\}$, $i = 0, 1, 2, 3$ and $k = 0, 1$;
- the 8 remaining K_4 of the partition of the K_{12} on the vertices α_i, $i \in \mathbb{Z}_4 \cup \left\{x_j^k,\ j \in \mathbb{Z}_8\right\}$, removing the K_4 $\{\alpha_0, \alpha_1, \alpha_2, \alpha_3\}$ and 4 C_3 $\left\{\alpha_i, x_{2i}^k, x_{2i+1}^k\right\}$, to obtain a total of 32 K_4;
- the 64 K_4 of the partition of the multipartite graph $K_{8\times 4}$ with vertex set $V_0 \cup V_1 \cup V_2 \cup V_3$.

Altogether the partition uses 8 $A_{6,5}$ and 97 K_4 and we have $A(6, 36) = 428$. □

The following corollary facilitates a kind of induction in general constructions.

Corollary 4.46. *When* $N = 36u + m$, $m = 0, 3, 6, 15, 21, 27, 33$, *and* $u \geq 4$, *then* $A(6, N) = 432u^2 + 24um - 4u + A(6, m)$.

Proof. From Theorem 4.21 there exists a 4-GDD of type $36^u m^1$ for each $u \geq 4$ and $m \in \{3, 6, 9, \ldots, 33\}$. Thus $A(6, 36u + m) \geq uA(6, 36) + \frac{4\cdot 36^2\cdot u(u-1)}{6\cdot 2} + \frac{4\cdot 36\cdot m\cdot u}{6} + A(6, m) = 432u^2 - 4u + 24um + A(6, m)$. □

We did not find decompositions for 18, 24, or 30 nor were we able to prove that the lower bound cannot be realized for those values.

For this reason, we need decompositions for larger values of N in order to compose them and obtain results for the whole congruence class (modulo 36). Moreover, since the bound cannot be realized for $N = 12$ we employ another result for the same class (see Theorem 4.51).

LEMMA 4.47. *The lower bound is realized for $N = 117$, i.e., $A(6, 117) = 4550$.*

Proof. The design is based on $\mathbb{Z}_{104}$ with 13 infinite points to be added. Consider the blocks

$$\mathcal{B}_1 = \{\{1,50,51,92\}, \{1,5,26,63\}, \{2,29,55,56\}, \{2,6,25,31\}, \{2,40,49,62\}, \{2,28,71,89\},$$
$$\{1,30,60,66\}, \{2,59,92,103\}, \{2,69,78,83\}, \{1,56,74,77\}, \{1,11,93,98\}\},$$

and $\mathcal{B}_2 = \{\{8,49,102\}, \{5,16,89\}, \{3,77,92\}, \{6,71,90\}, \{4,62,74\}, \{7,41,43\}, \{1,70,72\}, \{2,61,99\}\}$.

Each block in $\mathcal{B}_1$ generates 52 blocks, by adding $2a$ to each element for $a \in \mathbb{Z}_{52}$ and reducing modulo 104. The differences covered by $\mathcal{B}_1 \cup \mathcal{B}_2$ form the set $\mathbb{Z}_{104} \setminus (\{8a : a \in \mathbb{Z}_{13}\} \cup \{52\})$. To be precise, a difference d that occurs actually occurs twice, once in a pair $\{a, a+d\}$ with a even, and once in a pair $\{b, b+d\}$ with b odd, so that all 104 pairs in the cyclic orbit comprising the pairs of difference d arise once. Adding the block $\{0, 8a, 24a, 72a\}$ covers the differences $\{8a : a \in \mathbb{Z}_{13} \setminus \{0\}\}$, and 104 blocks are generated by adding each element of $\mathbb{Z}_{104}$ and reducing modulo 104. The blocks in $\mathcal{B}_2$ together contain 24 entries whose residues modulo 26 are $\mathbb{Z}_{26} \setminus \{0, 13\}$. The blocks $\{\{b_1 + 26x, b_2 + 26x, b_3 + 26x\} : \{b_1, b_2, b_3\} \in \mathcal{B}_2, x \in \mathbb{Z}_4\}$ form a partial parallel class missing the elements $\{13a : a \in \mathbb{Z}_8\}$. Now add the infinite point ∞_0 to each block of this partial parallel class to form $\mathcal{B}_{2,0}$. Form a new partial parallel class $\mathcal{B}_{2,a}$ for $1 \le a \le 12$ by adding $2a$ to each noninfinite point (modulo 104) and replacing ∞_0 by ∞_a. Now place a (13,4,1)-design on the 13 infinite points.

Finally, form 13 F_4 as follows. For $0 \le a < 13$, form an F_4 with center ∞_a and containing the triangles $\{\infty_a, a + 13x, a + 13x + 52\}$ for $x \in \mathbb{Z}_4$. □

LEMMA 4.48. *The lower bound is met with equality for $N = 7 \cdot 117 + m$ for $m \in \{3, 21, 27, 33\}$, i.e., for $N \in \{822, 840, 846, 852\}$.*

Proof. Form a 4-GDD of type $117^7 m^1$, and place a decomposition with cost $A(6, 117)$ on each of the seven groups of size 117 and a decomposition with cost $A(6, m)$ on the last. □

4.3. Optimal general constructions. The following three results give constructions that meet the lower bound. Therefore they determine the value of $A(6, N)$ for all values of N with few exceptions.

THEOREM 4.49. *The value of $A(6, N)$ for $N \equiv 1 \pmod 3$ is given by $A(6, N) = \lceil \frac{2R}{3} \rceil + \epsilon$, where $\epsilon = 2$ if $N \equiv 7$ or $10 \pmod{12}$ and 0 otherwise, except for $A(6, 7) = 17$, $A(6, 10) = 34$, $A(6, 19) = 119$.*

Proof. For $N \notin \{7, 10, 19\}$, by a result of Mills on covering K_N by K_4 (see Theorem 8.9 of [12]), there exists a covering of K_N with $\lceil \frac{N(N-1)}{12} \rceil$ K_4 and therefore $A = \lceil \frac{2R}{3} \rceil + \epsilon$. Lemmas 4.29, 4.32, and 4.38 give the result for the remaining values of N. □

THEOREM 4.50. *If $N \equiv 2 \pmod 3$, then $A(6, N) = \frac{2R+N+2}{3}$, except possibly for $N = 17$.*

Proof.

Case 1. $N \equiv 2$ or $11 \pmod{12}$.

To prove the theorem for $N \equiv 2$ or $11 \pmod{12}$, we show that K_N can be decomposed into one K_2, $\frac{N-2}{3}$ $A_{6,5}$ or $B_{6,5}$ and K_4.

- The result is true for $N = 2, 11, 14$ (Lemmas 4.33 and 4.35); for $N = 11$ the decomposition uses 1 K_2, 3 $A_{6,5}$, and 6 K_4; for $N = 14$ the decomposition uses 1 K_2, 4 $B_{6,5}$, and 11 K_4.
- If $N = 12u+2$, $u \geq 4$, then $K_{12u+2} - K_2$ can be decomposed into u $K_{14} - K_2$ and $K_{12\times u}$. Furthermore, each $K_{14} - K_2$ can be decomposed into 4 $B_{6,5}$ and 11 K_4 (Lemma 4.35), and $K_{12\times u}$ can be decomposed into $12u(u-1)$ K_4 (existence of a 4-GDD of type 12^u by Theorem 4.12).
- If $N = 12u + 11$, $u \geq 4$, then $K_{12u+11} - K_2$ can be decomposed into u $K_{14} - K_2$, one $K_{11} - K_2$, and $K_{12\times u,9}$. Furthermore $K_{14} - K_2$ and $K_{11} - K_2$ can be decomposed into $A_{6,5}$ or $B_{6,5}$ and K_4 (Lemmas 4.33 and 4.35), and $K_{12\times u,9}$ can be decomposed into K_4 (existence of a 4-GDD of type 12^u9 by Theorem 4.12).
- The theorem is also true for $N = 23, 26, 35$ by Lemmas 4.40, 4.41, and 4.44 and for $N = 38, 47$; for $N = 38$ (resp., 47), $K_{38} - K_2$ (resp., $K_{47} - K_2$) can be decomposed into four (resp., 5) $K_{11} - K_2$ plus $K_{9\times 4}$ (resp., $K_{9\times 5}$). Each $K_{11} - K_2$ can be decomposed into $A_{6,5}$ (resp., $B_{6,5}$) and K_4 (Lemmas 4.33 and 4.35), and $K_{9\times 4}$ (resp., $K_{9\times 5}$) can be decomposed into K_4 (existence of a 4-GDD of type 9^4 and 9^5).

Case 2. $N \equiv 5$ or $8 \pmod{12}$.

In this case, we prove that K_N can be decomposed into one C_4, $\frac{N-2}{3}$ $A_{6,5}$, or $B_{6,5}$ and K_4.

- That is true for $N = 5, 8$ (Lemmas 4.28 and 4.30); for $N = 5$, the decomposition uses one C_4 and one $A_{6,5}$; for $N = 8$ the decomposition uses one C_4, two $B_{6,5}$, and two K_4.
- If $N = 12u + 5$ (resp., $12u + 8$), $u \geq 4$, then K_N can be decomposed into u $K_{14}-K_2$, one K_5 (resp., K_8), and one $K_{12\times u,3}$ (resp., $K_{12\times u,6}$). Furthermore, each $K_{14} - K_2$ can itself be decomposed into $B_{6,5}$ and K_4, and $K_{12\times u,3}$ (resp., $K_{12\times u,6}$) into K_4 (existence of a 4-GDD of type 12^u3 and 12^u6).
- The theorem is also true for $N = 20, 29, 32$ by Lemmas 4.39, 4.42, and 4.43 and for $N = 41, 44$; for $N = 41$ (resp., 44), we use the decomposition of K_N into four $K_{11} - K_2$, one K_5 (resp., K_8), and $K_{9\times 4,3}$ (resp., $K_{9\times 4,6}$).
- It remains for us to solve the case $N = 17$. □

THEOREM 4.51. *If $N \equiv 0 \pmod 3$, then $A(6, N) = \lceil \frac{6R+2N}{9} \rceil + \epsilon$, where $\epsilon = 1$ if $N \equiv 18, 27 \pmod{36}$, and $\epsilon = 0$ otherwise, except for $N \in \{9, 12\}$ and possibly when*

$$\begin{aligned}
N &\equiv 0 \pmod{36} && \textit{and} && \lfloor N/36 \rfloor \in \{2,3\},\\
N &\equiv 3 \pmod{36} && \textit{and} && \lfloor N/36 \rfloor \in \{1,2,3\},\\
N &\equiv 6 \pmod{36} && \textit{and} && \lfloor N/36 \rfloor \in \{1,2,3\},\\
N &\equiv 9 \pmod{36} && \textit{and} && \lfloor N/36 \rfloor \in \{1,2,4,5,6,7,8,9,10\},\\
N &\equiv 12 \pmod{36} && \textit{and} && \lfloor N/36 \rfloor \leq 70, \neq 23,\\
N &\equiv 15 \pmod{36} && \textit{and} && \lfloor N/36 \rfloor \in \{1,2,3\},\\
N &\equiv 18 \pmod{36} && \textit{and} && \lfloor N/36 \rfloor \leq 70, \neq 23,\\
N &\equiv 21 \pmod{36} && \textit{and} && \lfloor N/36 \rfloor \in \{1,2,3\},\\
N &\equiv 24 \pmod{36} && \textit{and} && \lfloor N/36 \rfloor \leq 71, \neq 23,\\
N &\equiv 27 \pmod{36} && \textit{and} && \lfloor N/36 \rfloor \in \{1,2,3\},\\
N &\equiv 30 \pmod{36} && \textit{and} && \lfloor N/36 \rfloor \leq 68, \neq 22,\\
N &\equiv 33 \pmod{36} && \textit{and} && \lfloor N/36 \rfloor \in \{1,2,3\}.
\end{aligned}$$

Proof. First we treat cases when $N \equiv 0, 3, 6, 15, 21, 27, 33 \pmod{36}$. By Lemma 4.28 we have $A(6, 3) = 3$ and $A(6, 6) = 12$. By Corollary 4.37 we have $A(6, 15) = 74$, $A(6, 21) = 145$, $A(6, 27) = 241$, and $A(6, 33) = 360$. From Lemma 4.45 we have $A(6, 36) = 428$. Then applying Corollary 4.46 we have $A(6, 36u + m) = 432u^2 + 24um - 4u + A(6, m)$, $u \geq 4$, and $m = 0, 3, 6, 15, 21, 27, 33$.

To treat $N \equiv 9 \pmod{36}$, use Lemmas 4.47 and 4.45, together with a 4-GDD of type $36^u 117^1$ from Theorem 4.26 to establish that $A(6, 36u + 117) = u \cdot A(6, 36) + A(6, 117) + 432u(u - 1) + 2808u$ for $u \geq 8$.

Finally, to handle $N = 36u + m$ with $m \in \{822, 840, 846, 852\}$ (corresponding to cases when $N \equiv 30, 12, 18, 24 \pmod{36}$), use Lemmas 4.45 and 4.48, together with a 4-GDD of type $36^u m^1$ from Theorem 4.26 to obtain the result. The ingredient designs are available when $u \geq 47$ for $m = 822$, $u \geq 48$ for $m \in \{840, 846\}$, and $u \geq 49$ for $m = 852$. □

REFERENCES

[1] B. Beauquier, J.-C. Bermond, L. Gargano, P. Hell, S. Pérennes, and U. Vaccaro, *Graph problems arising from wavelength-routing in all-optical networks*, in IEEE Workshop on Optics and Computer Science, Geneva, Switzerland, 1997.

[2] F. E. Bennett, Y. Chang, G. Ge, and M. Greig, *Existence of* $(v, \{5, w^*\}, 1)$*-PBDs*, Discrete Math., 279 (2004), pp. 61–105.

[3] J.-C. Bermond and S. Ceroi, *Minimizing SONET ADMs in unidirectional WDM ring with grooming ratio* 3, Networks, 41 (2003), pp. 83–86.

[4] J.-C. Bermond, C. J. Colbourn, A. C. H. Ling, and M.-L. Yu, *Grooming in unidirectional rings:* $K_4 - e$ *designs*, Discrete Math., 284 (2004), pp. 67–72.

[5] J.-C. Bermond and D. Coudert, *Traffic grooming in unidirectional WDM ring networks using design theory*, in IEEE ICC, Anchorage, AK, 2003.

[6] J.-C. Bermond, D. Coudert, and X. Muñoz, *Traffic grooming in unidirectional WDM ring networks: The all-to-all unitary case*, in The 7th IFIP Working Conference on Optical Network Design & Modelling, 2003, pp. 1135–1153.

[7] J.-C. Bermond, C. Huang, A. Rosa, and D. Sotteau, *Decomposition of complete graphs into isomorphic subgraphs with five vertices*, Ars Combin., 10 (1980), pp. 211–254.

[8] J.-C. Bermond and D. Sotteau, *Graph decompositions and G-designs*, in 5th British Combinatorial Conference, Aberdeen, 1975, Congr. Numer. 15, Utilitas Math. Pub., Winnipeg, Canada, pp. 53–72.

[9] T. Beth, D. Jungnickel, and H. Lenz, *Design Theory*, Cambridge University Press, Cambridge, UK, 1993.

[10] R. Bose, S. Shrikhande, and E. Parker, *Further results on the construction of mutually orthogonal Latin squares and the falsity of Euler's conjecture*, Canad. J. Math., 12 (1960), pp. 189–203.

[11] A. L. Chiu and E. H. Modiano, *Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks*, IEEE/OSA J. Lightwave Tech., 18 (2000), pp. 2–12.

[12] C. J. Colbourn and J. H. Dinitz, eds., *The CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton, FL, 1996.

[13] C. J. Colbourn and A. C. H. Ling, *Wavelength add-drop multiplexing and minimizing SONET ADMs*, Discrete Math., 261 (2003), pp. 141–156.

[14] C. J. Colbourn and P.-J. Wan, *Minimizing drop cost for SONET/WDM networks with* $\frac{1}{8}$ *wavelength requirements*, Networks, 37 (2001), pp. 107–116.

[15] R. Dutta and N. Rouskas, *A survey of virtual topology design algorithms for wavelength routed optical networks*, Optical Networks, 1 (2000), pp. 73–89.

[16] R. Dutta and N. Rouskas, *On optimal traffic grooming in WDM rings*, IEEE J. Selected Areas Commun., 20 (2002), pp. 1–12.

[17] R. Dutta and N. Rouskas, *Traffic grooming in WDM networks: Past and future*, IEEE Network, 16 (2002), pp. 46–56.

[18] G. Ge and A. C. H. Ling, *Group divisible designs with block size four and group type* $g^u m^1$ *for small* g, Discrete Math., 285 (2004), pp. 97–120.

[19] G. Ge and A. C. H. Ling, *A survey on resolvable group divisible designs with block size four*, Discrete Math., 279 (2004), pp. 225–245.

[20] G. Ge and A. C. H. Ling, *Asymptotic results on the existence of 4-RGDDs and uniform 5-GDDs*, J. Combin. Designs, to appear.

[21] G. Ge and R. S. Rees, *On group-divisible designs with block size four and group-type $g^u m^1$*, Des. Codes Cryptogr., 27 (2002), pp. 5–24.

[22] G. Ge and R. S. Rees, *On group-divisible designs with block size four and group-type $6^u m^1$*, Discrete Math., 279 (2004), pp. 247–265.

[23] G. Ge, R. S. Rees, and L. Zhu, *Group-divisible designs with block size four and group-type $g^u m^1$ with m as large or as small as possible*, J. Combin. Theory Ser. A, 98 (2002), pp. 357–376.

[24] G. Ge and R. Wei, *HGDDs with block size four*, Discrete Math., 279 (2004), pp. 267–276.

[25] O. Gerstel, P. Lin, and G. Sasaki, *Wavelength assignment in a WDM ring to minimize cost of embedded SONET rings*, in IEEE Infocom, San Francisco, CA, 1998, pp. 94–101.

[26] O. Gerstel, R. Ramaswani, and G. Sasaki, *Cost-effective traffic grooming in WDM rings*, IEEE/ACM Trans. Networking, 8 (2000), pp. 618–630.

[27] O. Goldschmidt, D. Hochbaum, A. Levin, and E. Olinick, *The SONET edge-partition problem*, Networks, 41 (2003), pp. 13–23.

[28] J. Q. Hu, *Optimal traffic grooming for wavelength-division-multiplexing rings with all-to-all uniform traffic*, OSA J. Optical Networks, 1 (2002), pp. 32–42.

[29] J. Q. Hu, *Traffic grooming in WDM ring networks: A linear programming solution*, OSA J. Optical Networks, 1 (2002), pp. 397–408.

[30] D. L. Kreher and D. R. Stinson, *Small group divisible designs with block size* 4, J. Statist. Plann. Inference, 58 (1997), pp. 111–118.

[31] E. R. Lamken and R. M. Wilson, *Decompositions of edge-colored complete graphs*, J. Combin. Theory Ser. A, 89 (2000), pp. 149–200.

[32] E. Modiano and P. Lin, *Traffic grooming in WDM networks*, IEEE Commun. Magazine, 39 (2001), pp. 124–129.

[33] R. S. Rees, *Group-divisible designs with block size k having $k+1$ groups for $k = 4, 5$*, J. Combin. Designs, 8 (2000), pp. 363–386.

[34] A. Somani, *Survivable traffic grooming in WDM networks*, in Broad Band Optical Fiber Communications Technology, D. K. Gautam, ed., Nirtali Prakashan, Pune, India, 2001, pp. 17–45.

[35] D. R. Stinson, *Determination of a packing number*, Ars Combin., 3 (1977), pp. 89–114.

[36] P.-J. Wan, G. Calinescu, L. Liu, and O. Frieder, *Grooming of arbitrary traffic in SONET/WDM BLSRs*, IEEE J. Selected Areas Commun., 18 (2000), pp. 1995–2003.

[37] J. Wang, W. Cho, V. Vemuri, and B. Mukherjee, *Improved approaches for cost-effective traffic grooming in WDM ring networks: Ilp formulations and single-hop and multihop connections*, IEEE/OSA J. Lightwave Technology, 19 (2001), pp. 1645–1653.

[38] R. M. Wilson, *Decomposition of complete graphs into subgraphs isomorphic to a given graph*, in Congr. Numer., 15, 1976, pp. 647–659.

[39] X. Yuan and A. Fulay, *Wavelength assignment to minimize the number of SONET ADMs in WDM rings*, in IEEE International Conference on Communications, New York, 2002.

[40] X. Zhang and C. Qiao, *An effective and comprehensive approach for traffic grooming and wavelength assignment in SONET/WDM rings*, IEEE/ACM Trans. Networking, 8 (2000), pp. 608–617.

SIAM J. Discrete Math.
Vol. 19, No. 3, pp. 543–552

DISKS ON A TREE: ANALYSIS OF A COMBINATORIAL GAME[*]

TOMÁS FEDER[†] AND CARLOS SUBI[‡]

Abstract. Anderson et al. [*Amer. Math. Monthly*, 96 (1989), pp. 481–493] studied a combinatorial game on an infinite path that is started with n disks at a vertex and ends with the disks spread between $k = \lfloor n/2 \rfloor$ vertices to the left and to the right of the initial vertex. They showed that the number of steps the game takes to converge to the final configuration is $ck^2 + o(k^2)$ for some constant c. We generalize this game to the case of an infinite rooted tree, where each vertex has degree $d+1$ and where the earlier game corresponds to the case $d = 1$. We determine the final configuration when the game is started with n disks at the root and show that in this final configuration all disks are at depth at most $k = \Theta(\log_d n)$ for $d \geq 2$. We also show that the number of steps that the game takes to converge to the final configuration in this case is at most $O(k(1+\log_d k))$, so that the convergence is faster than what it was for the case $d = 1$. We generalize the game to the case where the vertices at depth i in the tree have $d_i \geq 2$ children, where the d_i are not necessarily the same, and show that the convergence time in this case is at most $O(k^{1.5} + k \log_{d_{\min}} d_{\max})$, where $d_{\min}$ and $d_{\max}$ are the smallest and largest d_i, respectively.

Key words. disks, tree, chip-firing games

AMS subject classifications. 05-99

DOI. 10.1137/S0895480102418002

1. Introduction.

In this article we study a very simple combinatorial game that can be played with several piles of disks arranged in a tree. At each unit of time, each pile, sitting at a vertex of degree d, is divided into d equal piles, which are moved to the d neighbors of the vertex, leaving a remainder of at most $d-1$ disks at the original vertex.

We are interested in the case of an infinite rooted tree, where the root is at depth zero and has $d_0 + 1$ children, and in general each vertex at depth i has d_i children. The initial configuration has n disks.

This game was studied by Anderson et al. [1] in the case where all $d_i = 1$, so that the tree is an infinite path. They determined the final configuration and showed that this configuration is reached in $c(n/2)^2 + o(n^2)$ steps for some $1/3 \leq c \leq \pi^2/6 - 1$. Björner, Lovász, and Shor [3] studied the related slowed-down game on an arbitary graph with n vertices and m edges, where a single move consists of selecting a vertex of degree d with at least d disks and moving these disks to the d neighbors. They showed that the final configuration and the number of moves depend only on the initial configuration and that the game is infinite if the number of disks is greater than $2m - n$, is finite if the number of disks is smaller than m, and can be finite or infinite depending on the initial configuration if the number of disks is between m and $2m - n$. Tardos [27] showed that there exist graphs with an initial configuration for which the number of steps of this slowed-down game is $\Omega(n^4)$ and that the number of steps is always bounded by $2nmd = O(n^4)$, where d is the diameter of the graph.

Various versions of such games on graphs have been studied as chip-firing games and Abelian sandpile models, including the work of Goles et al. [20, 21, 22, 23, 24, 25, 26], Dhar et al. [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19], and others [2, 4].

[*] Received by the editors November 15, 2002; accepted for publication (in revised form) January 13, 2005; published electronically November 15, 2005.
http://www.siam.org/journals/sidma/19-3/41800.html

[†] 268 Waverley St., Palo Alto, CA 94301 (tomas@theory.stanford.edu).

[‡] 738 Cowper St., Palo Alto, CA 94301 (carlos_subi@hotmail.com).

Back at the case of an infinite tree, if we denote by x_{ri} the number of disks at any one vertex at depth i after r steps, then we obtain the recurrence $x_{r0} = x_{(r-1)0} \bmod (d_0+1) + (d_0+1)\lfloor x_{(r-1)1}/(d_1+1)\rfloor$, and for $i \geq 1$, $x_{ri} = \lfloor x_{(r-1)(i-1)}/(d_{i-1}+1)\rfloor + x_{(r-1)i} \bmod (d_i+1) + d_i\lfloor x_{(r-1)(i+1)}/(d_{i+1}+1)\rfloor$. The base case is $x_{00} = n$ and $x_{0i} = 0$ for $i \geq 1$.

We first determine the final configuration for this game. For this final configuration, we denote by n_i the number of disks sitting at a subtree rooted at a vertex at depth i and by e_i the number of disks sitting at a vertex at depth i. Then $n_0 = n$, $e_0 = n \bmod (d_0+1)$, $n_1 = \lfloor n/(d_0+1)\rfloor$, $e_i = 1 + (n_i - 1) \bmod d_i$ for $1 \leq i \leq k$, and $n_{i+1} = \lfloor (n_i-1)/d_i\rfloor$ for $1 \leq i \leq k$. Here k is the first i such that $n_{i+1} = 0$. In fact no disk ever reaches depth $k+1$.

We next study the number of steps it takes for the game to reach its final configuration. We focus on the case where all $d_i = d \geq 2$. We first consider the special case where $n = (d+1)(d^k-1)/(d-1)$, so that all $e_i = 1$ for $1 \leq i \leq k$. For this case, we show that the number of steps is bounded by $O(k) = O(\log_d n)$. The proof is based on a comparison with the fractional game where no remainder is left at a vertex.

For general n, repeated applications of the preceding result give a bound of $O(k(1+\log_d k)) = O(\log_d n \log_d \log_d n)$ on the number of steps.

We next consider the case where all $d_i \geq 2$ are not necessarily equal. Again we first consider the special case where all $e_i = 1$ for $1 \leq i \leq k$. For this case, we show that the number of steps is bounded by $O(\log_{d_{\min}} n) = O(\sum_{i\leq k} \log_{d_{\min}} d_i) = O(k \log_{d_{\min}} d_{\max})$, where $d_{\min}$ and $d_{\max}$ denote the smallest and the largest d_i for $1 \leq i \leq k$.

We then obtain bounds for general n. If $2 \leq d_i \leq d_j$ for $1 \leq i \leq j$, then the number of steps is bounded by $O(k \log_{d_{\min}}(kd_{\max}))$.

If $d_i \geq d_j$ for $1 \leq i \leq j$, then the number of steps is at most $2k^2$.

In the general case of n arbitrary and all $d_i \geq 2$ for $i \geq 1$, the number of steps is bounded by $O(k^{1.5} + \log_{d_{\min}} n) = O(k^{1.5} + k\log_{d_{\min}} d_{\max})$.

We finally obtain a lower bound of $\Omega(k + \max_{1\leq i\leq k} \sum_{i\leq j\leq k} \log_{d_i} d_j)$ on the number of steps if all $d_i \geq 2$. Thus the upper bound for the case where all $e_i = 1$ for $0 \leq i \leq k$ is tight up to constant factors, provided $\log_{d_{\min}} d_1 = O(1)$, that is, $d_1 \leq d_{\min}^{O(1)}$.

The analysis of the game in the case where some of the d_i for $i \geq 1$ satisfy $d_i = 1$ and some satisfy $d_i \geq 2$ remains open.

The case of a tree is thus interesting because, unlike the case of a path, the number of steps depends only logarithmically on the number of disks, and the dependence seems to be essentially linear in the depth of the tree reached by the final configuration. This contrasts with the fact that in the case of a path, the dependence is quadratic in the length of the path, which equals in that case the number of initial disks. In fact, none of the previously studied cases in the literature shows dependence that is only logarithmic in the number of disks or linear in the diameter of the graph.

2. The final configuration.

Recall that if we denote by x_{ri} the number of disks at any one vertex at depth i after r steps, then we obtain the recurrence $x_{r0} = x_{(r-1)0} \bmod (d_0+1) + (d_0+1)\lfloor x_{(r-1)1}/(d_1+1)\rfloor$, and for $i \geq 1$, $x_{ri} = \lfloor x_{(r-1)(i-1)}/(d_{i-1}+1)\rfloor + x_{(r-1)i} \bmod (d_i+1) + d_i\lfloor x_{(r-1)(i+1)}/(d_{i+1}+1)\rfloor$. The base case is $x_{00} = n$ and $x_{0i} = 0$ for $i \geq 1$.

LEMMA 1. *The combinatorial game terminates in some final configuration.*

Proof. Consider the potential function $\phi_r = \sum_i t_{ri} i^2$, where t_{ri} is the total number of disks at all vertices at depth i after r steps. This potential function increases at each step: if d_0+1 disks at depth 0 are moved to depth 1, it increases by d_0+1; if

d_i+1 disks at depth i are moved so that one disk goes to depth $i-1$ and d_i disks go to depth $i+1$, it increases by $(i-1)^2+d_i(i+1)^2-(d_i+1)i^2=2(d_i-1)i+d_i+1$.

Consequently no configuration ever repeats. Suppose that after some number r of steps, all depths up to depth k_r have been reached by disks. The number of disks at depth r cannot be more than $n/2^{k_r}\geq 1$. Therefore $k_r\leq \log_2 n$.

Since there is a finite number of configurations that never reach depth $2n$, and no configuration ever repeats because the potential function always increases, a final configuration must eventually be reached. □

We determine the final configuration for this game. For this final configuration, we denote by n_i the number of disks sitting at a subtree rooted at a vertex a depth i and by e_i the number of disks sitting at a vertex at depth i.

Theorem 1. *The final configuration is given by $n_0=n$, $e_0=n \bmod (d_0+1)$, $n_1=\lfloor n/(d_0+1)\rfloor$, $e_i=1+(n_i-1) \bmod d_i$ for $1\leq i\leq k$, and $n_{i+1}=\lfloor (n_i-1)/d_i\rfloor$ for $1\leq i\leq k$. Here k is the first i such that $n_{i+1}=0$. In fact no disk ever reaches depth $k+1$.*

Proof. Since the d_0+1 subtrees rooted at depth 1 are identical, it follows that the total number of disks remaining in such subtrees is a multiple of d_0+1. Since $0\leq e_0\leq d_0$, it follows that $e_0=n \bmod (d_0+1)$, and therefore $n_1=\lfloor n/(d_0+1)\rfloor$.

Consider the n_i disks remaining at a subtree rooted at depth i. Since a vertex at depth i has d_i identical subtrees rooted at depth $i+1$, it follows that the total number of disks remaining in such subtrees is a multiple of d_i. Since $0\leq e_i\leq d_i$, it follows that $e_i=n_i \bmod d_i$ if n_i is not divisible by d_i, and otherwise either $e_i=0$ or $e_i=d_i$. We shall show that $e_i=0$ is not possible, so in this case $e_i=d_i$, and so in general $e_i=1+(n_i-1) \bmod d_i$, implying $n_{i+1}=\lfloor (n_i-1)/d_i\rfloor$ for $i\geq 2$. Since $n_{i+1}=0$, no disk ever reaches depth $k+1$.

It remains to show that $e_i=0$ is not possible for $1\leq i\leq k$. Suppose $e_i=0$. The last time disks left depth i, each vertex at depth $i-1$ received at least d_{i-1} disks from its children, so $e_{i-1}=d_{i-1}$. Similarly, there were 0 disks at depth $i-1$ before these d_{i-1} disks arrived from depth i; otherwise we would later get a nonzero number of disks at depth i, so the last time disks left depth $i-1$ happened before, and each vertex at depth $i-2$ received at least d_{i-2} disks from its children, so $e_{i-2}=d_{i-2}$. Proceeding inductively, we obtain $e_1=d_1$, and there were 0 disks at depth 1 before these d_1 disks arrived from depth 2, so the last time disks left depth 1, the root at depth 0 received at least d_0+1 disks from its children. This would give $e_0\geq d_0+1$, contrary to the fact that $e_0\leq d_0$. This completes the proof. □

3. The case of all $d_i=d\geq 2$.

We shall study the number of steps it takes for the game to reach its final configuration. In this section, all d_i have the same value $d_i=d\geq 2$. If we denote by x_{ri} the number of disks at a vertex at depth i after r steps, then we obtain the recurrence $x_{r0}=x_{(r-1)0} \bmod (d+1)+(d+1)\lfloor x_{(r-1)1}/(d+1)\rfloor$, and for $i\geq 1$, $x_{ri}=\lfloor x_{(r-1)(i-1)}/(d+1)\rfloor+x_{(r-1)i} \bmod (d+1)+d\lfloor x_{(r-1)(i+1)}/(d+1)\rfloor$. The base case is $x_{00}=n$ and $x_{0i}=0$ for $i\geq 1$.

There is a closely related fractional game where no remainder is left at a vertex. For this fractional game, we study the recurrence $y_{ri}=y_{(r-1)(i-1)}/(d+1)+dy_{(r-1)(i+1)}/(d+1)$. The base case is $y_{00}=n$ and $y_{0i}=0$ for $i\neq 0$. Here we are allowing i to be negative.

Lemma 2. *The solution of the recurrence is $y_{r(2i-r)}=n(1/d)^i(d/(d+1))^r\binom{r}{i}$ and $y_{ri}=0$ for $i+r$ odd.*

Proof. Clearly $y_{ri}=0$ unless r and i are either both even or both odd. Let $z_{r(2i-r)}=d^iy_{r(2i-r)}$. Then $z_{(r+1)(2i-(r+1))}/d=z_{r(2(i-1)-r)}/(d+1)+z_{r(2i-r)}/(d+1)$.

Let $w_{r(2i-r)} = ((d+1)/d)^r z_{r(2i-r)}$. Then $w_{(r+1)(2i-(r+1))} = w_{r(2(i-1)-r)} + w_{r(2i-r)}$.

Then $w_{r(2i-r)} = n\binom{r}{i}$. Therefore $z_{r(2i-r)} = n(d/(d+1))^r\binom{r}{i}$, and so $y_{r(2i-r)} = n(1/d)^i(d/(d+1))^r\binom{r}{i}$. □

We shall use the concept of slowed-down versions of the combinatorial game. Here not all disks that could be moved at a given point in time are moved, so that moving these disks is delayed until later. This means that the slowed-down game takes longer to reach the final configuration than the original game. See also [3, 27]. Thus in a slowed-down game, we still move the same number of disks to each neighbor, but we may choose a smaller number of such disks to move, so that a number larger than the smallest possible remainder is left at each chosen vertex. This results in partially postponing the full move that would happen at a step, so that the rest of the move will happen later. The result is that the number of steps is increased when we go to the slowed-down game, yet the same final configuration is eventually reached. We also consider at times a fractional game, where fractions of disks may be moved to all neighbors, in the same quantity to each neighbor, as opposed to moving only full disks, which results again in postponing the move of the remaining fraction, while eventually reaching the same final configuration.

LEMMA 3. *In the combinatorial game with all $d_i = d \geq 2$ and $n = (d+1)(d^k - 1)/(d-1)$, so that $e_0 = 0$ and all $e_i = 1$ for $1 \leq i \leq k$, depth k is reached in $O(k)$ steps (independently of d).*

Proof. We slow down the combinatorial game by requiring that if there are at least d disks at a vertex at depth i after $r-1$ steps, then exactly d disks are left at depth i for the rth step; if there are at most d disks at a vertex at depth i, then none of these disks is moved. We show that this slowed-down fractional game reaches depth k within $O(k)$ steps. This implies that the original combinatorial game, which is not slowed down, will reach depth k as well.

The numbers of disks t_{ri} for the slowed down game are upper bounded by $t_{ri} \leq d + y_{ri}$, where the y_{ri} are the quantities from the recurrence for the preceding lemma, since disks in excess of d are moved according to fractional game defining the y_{ri}, and so the claim follows by induction. That is, the game played above d disks always has $t_{ri} - d \leq y_{ri}$, since those excess disks satisfy the recurrence for the y_{ri}, except that some disks may be lost if they reach a pile with fewer than d disks.

We bound the $y_{r(2i-r)}$ for $i \geq r/2$ by

$$y_{r(2i-r)} \leq n(1/d)^i(2d/(d+1))^r \leq n(4d/(d+1)^2)^{r/2}.$$

If we let $r = ck$ for a large constant c, then for $i \geq r/2$ we have $y_{r(2i-r)} \leq n(1/d)^{c'k}$ for another large constant c' depending on c.

If all vertices at depth $0 \leq i \leq k-1$ have d disks, then this accounts for exactly $n-1$ disks. The excess $y_{r(2i-r)} \leq n(1/d)^{c'k}$ in the bound $t_{ri} \leq d + y_{ri}$ for $0 \leq i \leq k-1$ accounts for strictly less than 1 disk if c' is large enough. Therefore some fraction of one disk must have reached depth k by step $r = ck$ in the slowed-down fractional game, so at least one disk will have reached depth k by step $r = ck$ in the combinatorial game. □

Define a *special configuration* to be a configuration where the sequence $x_{r0}x_{r1}\cdots x_{rk}$ is given by $01^*((d+1)d^*01^*)^*1$ or by $((d+1)d^*01^*)^+1$. Here x^* denotes any nonnegative number of copies of x, and x^+ denotes any positive number of copies of x.

LEMMA 4. *In the combinatorial game with all $d_i = d \geq 2$ and $n = (d+1)(d^k - 1)/(d-1)$, so that $e_0 = 0$ and all $e_i = 1$ for $1 \leq i \leq k$, after depth k is reached, we have a special configuration.*

Proof. After depth k is reached, there will be 1 disk at each vertex at depth k. A vertex at depth $k-1$ has at most 1 disk by a count on the total number of disks. If there is 1 disk at depth $k-1$, then we proceed inductively on k. If there are 0 disks at depth $k-1$, then the last time disks were moved from depth $k-1$ we obtained at least d disks at a vertex at depth $k-2$ and at most $d+1$ disks at such a vertex by a count on the number of disks. If there are exactly d disks, then again the last time disks were moved from depth $k-2$ we obtained at least d disks at a vertex at depth $k-3$, and so on. This accounts for the sequence ending in $((d+1)d^*01^*)1$. The number of disks accounted by such a sequence is the same as for a sequence of the same length of the form 1^*, so we may again proceed inductively to obtain again a sequence ending in $((d+1)d^*01^*)^2 1$, and so on for a sequence ending in $((d+1)d^*01^*)^* 1$. The resulting number of disks for the root at depth 0 will be either $d+1$ or 0, giving one of the two kinds of special configuration. ◻

LEMMA 5. *A special configuration with $k \geq 2$ takes at most $2k-3$ steps to reach the configuration 01^* with $e_0 = 0$ and $e_i = 1$ for $1 \leq i \leq k$.*

Proof. We show that each step decreases by at least 1 the number of $x_i = d$, which is at most $k-2$, in some slowed-down game. To see this, if some d is preceded by a 1, then we must in particular have a subsequence $1(d+1)(0(d+1))^r d$ for some r, which gives rise in one step to the subsequence $(d+1)(0(d+1))^{r+1}$, decreasing the number of d's by 1. If the first d is not preceded by a 1, then the initial sequence is either $(0(d+1))^r d$, giving in one step $(d+1)(0(d+1))^r$, or $(d+1)(0(d+1))^r d$, giving in one step $(0(d+1))^{r+1}$, again decreasing the number of d's by 1.

Once there remain no d's, each step increases the number of 1's at the end by 1, since the sequence must be of one of the two forms $01^*((d+1)01^*)^*1$, or by $((d+1)01^*)^+1$. It thus takes at most $k-1$ steps to reach 01^* for a total of $(k-2)+(k-1) = 2k-3$ steps. ◻

Combining Lemmas 3, 4, and 5, we have that the combinatorial game takes $O(k)$ steps to reach depth k by Lemma 3, at which point we have a special configuration by Lemma 4, and the remaining steps that take this special configuration to a final configuration are bounded in a slowed-down game analysis of these remaining steps by $2k-3$, for a total of $O(k)$ steps. We thus obtain the following.

THEOREM 2. *In the combinatorial game with all $d_i = d \geq 2$ and $n = (d+1)(d^k - 1)/(d-1)$, so that $e_0 = 0$ and all $e_i = 1$ for $1 \leq i \leq k$, it takes $O(k)$ steps to reach the final configuration, independent of d.*

For the rest of the section, it will be convenient to change the value of d_0. This will be justified by the following.

LEMMA 6. *The combinatorial game with n disks and some value of d_0 is equivalent to the game with $\lfloor n/(d_0+1) \rfloor$ disks on a tree modified to have a degree 1 root; that is, both games take the same number of steps. Thus there is a correspondence between different possible values of d_0 via the value $d_0 = 0$.*

Proof. In both games, the first step moves $\lfloor n/(d_0+1) \rfloor$ disks from the root at depth 0 to each vertex at depth 1. In subsequent pairs of steps $2i$ and $2i+1$, if the root receives r disks from each vertex at depth 1 in step $2i$, then it sends r disks back to each vertex at depth 1 in step $2i+1$. ◻

Assume still that all $d_i = d \geq 2$ for $1 \leq i \leq k$ but set $d_0 = d-2$.

Lemma 7. *A slowed-down game reaches a configuration with $x_{ri} \leq (d-1)(k+1)$ and $x_{ri} \geq x_{rj}$ for $i \leq j$ in $r = O(k)$ steps.*

Proof. Repeatedly subtract the largest $n' \leq n$ from n that can be replaced by a sequence of the form 1^l with $l \leq k$ by the result of Theorem 2, in $O(k)$ steps. Each value of l will be chosen at most $d-1$ times, since the sequence d^l would give instead the sequence 1^{l+1}. Notice that this takes a total of $O(k)$ steps, since a slowed-down game can simultaneously carry out the different steps that lead to each 1^l.

The result, after $O(k)$ steps of this slowed-down game, is thus at most $k+1$ sequences s_l^l with $0 \leq s_l \leq d-1$, for $0 \leq l \leq k$, and these sequences together prove the lemma. □

Lemma 8. *In a slowed-down game, a configuration with $x_{ri} \leq (d-1)(k+1)$ and $x_{ri} \geq x_{rj}$ for $i \leq j$ leads to a configuration with $x_{ri} = O(d(1+\log_d k))$ in $O(k(1+\log_d k))$ steps.*

Proof. Subtract from each x_{ri} at most d elements so that each x_{ri} is a multiple of d. Now decompose the configuration of resulting x'_{ri} into sequences of the form d^l, and replace each such sequence by a sequence of the form 1^{l+1} in at most $2k$ steps by an application of Lemma 5. This reduces the largest x_{ri} by a factor of d.

Performing this transformation $O(1+\log_d k)$ times, we will be left just with the $O(1+\log_d k)$ remainders of at most d elements for $x_{r'i}$, so that $x_{r'i} = O(d(1+\log_d k))$ after $O(k(1+\log_d k))$ steps. □

Lemma 9. *In a slowed-down game, a configuration with $x_{ri} = O(d(1+\log_d k))$ leads to the final configuration in $O(k(1+\log_d k))$ steps.*

Proof. There exists a special configuration $v_i \leq x_{ri}$ such that $v_i = d$ or $v_i = d+1$ whenever $x_{ri} \geq d+1$, and if $v_i = 0$, then $x_{ri} \leq d-1$. To see this, replace any sequence of entries x_{ri} that are at least $d+1$ by a sequence $(d+1)d^l$ of v_{ri}, adding some extra v_i set to d at the end for x_{ri} that are equal to d as well. Insert in between blocks of the form 01^l, noting that each v_i set to 0 will then correspond to x_{ri} that are at most $d-1$, since otherwise a d would have been used for v_i. This gives a special configuration.

Such a special configuration of v_i leads in $2k$ steps to a sequence of the form 01^l by Lemma 5, thus reducing the largest x_{ri} by at least $d-1$.

Performing this transformation $O(1+\log_d k))$ times will ensure that all resulting x_{ri} have value at most d, and we thus have a final configuration in $O(k(1+\log_d k))$ steps. □

Combining Lemmas 6, 7, 8, and 9, we obtain the following.

Theorem 3. *In the combinatorial game with all $d_i = d \geq 2$ and arbitrary n, it takes $O(k(1+\log_d k))$ steps to reach the final configuration.*

4. The case of arbitrary $d_i \geq 2$.

In this section, the d_i may have different values, but all $d_i \geq 2$ for $1 \leq i \leq k$. Recall that if we denote by x_{ri} the number of disks at a vertex at depth i after r steps, then we obtain the recurrence $x_{r0} = x_{(r-1)0} \bmod (d_0+1) + (d_0+1)\lfloor x_{(r-1)1}/(d_1+1)\rfloor$, and for $i \geq 1$, $x_{ri} = \lfloor x_{(r-1)(i-1)}/(d_{i-1}+1)\rfloor + x_{(r-1)i} \bmod (d_i+1) + d_i \lfloor x_{(r-1)(i+1)}/(d_{i+1}+1)\rfloor$. The base case is $x_{00} = n$ and $x_{0i} = 0$ for $i \geq 1$.

Let $d = d_{\min}$ denote the mininum d_i for $1 \leq i \leq k$. By Lemma 6, we may assume $d_0 = d_{\min}$. We again define a closely related fractional game with no remainders, with recurrence $s_{ri} = s_{(r-1)(i-1)}/(d_{i-1}+1) + d_i s_{(r-1)(i+1)}/(d_{i+1}+1)$ for $i \geq 1$, $s_{r0} = (d_0+1)s_{(r-1)1}/(d_1+1)$. The base case is $s_{00} = n$, $s_{0i} = 0$ for $i \geq 1$.

Lemma 10. *The solution of the recurrence has*

$$s_{r(2i-r)} \leq n(d_{2i-r}+1)(1/d)^i (d/(d+1))^r \binom{r}{i}$$

for $i \geq r/2$; otherwise $u_{ri} = 0$.

Proof. Define $u_{ri} = s_{ri}/(d_i+1)$. We obtain the recurrence $u_{ri} = u_{(r-1)(i-1)}/(d_i + 1) + d_i u_{(r-1)(i+1)}/(d_i+1)$ for $i \geq 1$, $u_{r0} = u_{(r-1)1}$. Setting $d = d_{\min}$, it suffices to show $u_{r(2i-r)} \leq y_{r(2i-r)}$, with y_{ri} given as in Lemma 2.

We show this by induction on r. If $i > (r+1)/2$, then $u_{(r+1)(2i-(r+1))} = u_{r(2(i-1)-r)}/(d_{2i-(r+1)}+1) + d_{2i-(r+1)} u_{r(2i-r)}/(d_{2i-(r+1)}+1) \leq y_{r(2(i-1)-r)}/(d_{2i-(r+1)} + 1) + d_{2i-(r+1)} y_{r(2i-r)}/(d_{2i-(r+1)}+1) \leq y_{r(2(i-1)-r)}/(d+1) + d y_{r(2i-r)}/(d+1) = y_{(r+1)(2i-(r+1))}$, since $y_{r(2(i-1)-r)} \geq y_{r(2i-r)}$ by Lemma 2. If $i = (r+1)/2$, then $u_{(r+1)(2i-(r+1))} = u_{r(2i-r)} \leq y_{r(2i-r)} \leq y_{(r+1)(2i-(r+1))}$. □

Lemma 11. *In the combinatorial game with all $d_i \geq 2$, and with $e_0 = 0$ and all $e_i = 1$ for $1 \leq i \leq k$, depth k is reached in $O(\log_{d_{\min}} n)$ steps.*

Proof. The proof is similar to that of Lemma 3. We slow down the combinatorial game by requiring that if there are at least d_i disks at a vertex at depth i after $r-1$ steps, then exactly d_i disks are left at depth i for the rth step; if there are at most d_i disks at a vertex at depth i, then none of these disks is moved. We show that this slowed-down fractional game reaches depth k within $O(k)$ steps. This implies that the original combinatorial game, which is not slowed down, will reach depth k as well.

The number of disks t_{ri} for the slowed-down game are upper bounded by $t_{ri} \leq d_i + s_{ri}$, where the s_{ri} are the quantities from the recurrence from the preceding lemma, since disks in excess of d are moved according to the fractional game defining the s_{ri}.

We bound the $s_{r(2i-r)}$ for $i \geq r/2$ by $s_{s(2i-r)} \leq n(d_{2i-r}+1)(4d/(d+1)^2)^{r/2}$ for $d = d_{\min}$. If we let $r = c \log_d n$ for a large constant c, then for $i \geq r/2$, we have $s_{r(2i-r)} \leq (1/n)^{c'}$ for another large constant c'.

If all vertices at depth $0 \leq i \leq k-1$ have d_i disks, then this accounts for exactly $n-1$ disks. The excess $s_{r(2i-r)} \leq (1/n)^{c'}$ in the bound $t_{ri} \leq d_i + s_{ri}$ for $0 \leq i \leq k-1$ accounts for strictly less than 1 disk if c' is large enough. Therefore some fraction of one disk must have reached depth k by step $r = c \log_d n$ in the slowed-down fractional game, so at least one disk will have reached depth k by step $r = c \log_d n$ in the combinatorial game. □

Define a *special configuration* to be a configuration where the sequence $x_{r0} x_{r1} \cdots x_{rk}$ is given by $01^*((d_i+1)d_i^* 01^*)^* 1$ or by $((d_i+1)d_i^* 01^*)^+ 1$, where the corresponding d_i is chosen for position x_{ri}. The arguments of Lemmas 4 and 5 yield the following two lemmas.

Lemma 12. *In the combinatorial game with all $d_i \geq 2$ with $e_0 = 0$ and all $e_i = 1$ for $1 \leq i \leq k$, after depth k is reached, we have a special configuration.*

Lemma 13. *A special configuration with $k \geq 2$ takes at most $2k-3$ steps to reach the configuration 01^* with $e_0 = 0$ and $e_i = 1$ for $1 \leq i \leq k$.*

Combining Lemmas 11, 12, and 13 yields the following.

Theorem 4. *In the combinatorial game with all $d_i \geq 2$, and with $e_0 = 0$ and all $e_i = 1$ for $1 \leq i \leq k$, it takes $O(\log_{d_{\min}} n) = O(\sum_{i \leq k} \log_{d_{\min}} d_i) = O(k \log_{d_{\min}} d_{\max})$ steps to reach the final configuration, where $d_{\min}$ and $d_{\max}$ denote the smallest and the largest d_i for $1 \leq i \leq k$.*

We now consider cases with arbitrary n. By Lemma 6, we may set $d_0 = d_{\min} - 2$.

Lemma 14. *A slowed-down game reaches a configuration with $x_{ri} \leq (d_{\max} - 1)(k+1)$ and $x_{ri} \geq x_{rj}$ for $i \geq j$ in $O(\log_{d_{\min}} n)$ steps.*

Proof. The proof is as in Lemma 7. Repeatedly subtract the largest $n' \leq n$ from n that can be replaced by a sequence of the form 1^l with $r \leq k$ by the result of Theorem 4 in $O(\log_{d_{\min}} n)$ steps. Each value of l will be chosen at most $d_{\max} - 1$ times, since the sequence $(d_0+2)d_1 \cdots d_l$ would give instead the sequence 1^{l+1}.

The result, after $O(\log_{d_{\min}} n)$ steps of this slowed-down game, is thus at most $k+1$ sequences s_l^l with $0 \leq s_l \leq d_{\max} - 1$, for $0 \leq l \leq k$, and these sequences together prove the lemma. □

Lemma 15. *Suppose $2 \leq d_i \leq d_j$ for $1 \leq i \leq j$. A configuration with $x_{ri} \leq (d_{\max} - 1)(k+1)$ and $x_{ri} \geq x_{rj}$ for $i \leq j$ leads to a configuration with $x_{r'i} = O((d_i+1)\log_{d_{\min}}(kd_{\max}))$ in $O(k \log_{d_{\min}}(kd_{\max}))$ steps.*

Proof. The proof is as in Lemma 8. Subtract from each x_{ri} at most d_i elements so that each x_{ri} is a multiple of d_i; for x_{r0}, subtract at most d_0+1 elements so that x_{r0} is a multiple of d_0+2. Note that if $x_{ri} < d_i$, then $x_{r(i+1)} < d_{i+1}$, since $x_{r(i+1)} \leq x_{ri}$ and $d_i \leq d_{i+1}$. Now decompose the configuration of resulting x'_{ri} into sequences of the form $(d_0+2)d_1 \cdots d_l$ and replace each such sequence by a sequence of the form 1^{l+1} in at most $2k$ steps by an application of Lemma 13. This reduces the largest x_{ri} by a factor of $d_{\min}$.

Performing this transformation $O(\log_{d_{\min}}(kd_{\max}))$ times, we will be left just with the $O(\log_{d_{\min}}(kd_{\max}))$ remainders of at most d_i elements for $x_{r'i}$ or d_0+1 elements for $x_{r'0}$, so that $x_{r'i} = O((d_i+1)\log_{d_{\min}}(kd_{\max}))$ after $O(k\log_{d_{\min}}(kd_{\max}))$ steps. □

Lemma 16. *For any value V, a configuration with $x_{r'i} = O((d_i+1)V)$ leads to the final configuration in $O(kV)$ steps.*

Proof. The proof is as in Lemma 9. There exists a special configuration $v_i \leq x_{ri}$ such that $v_i = d_i$ or $v_i = d_i + 1$ whenever $x_{ri} \geq d_i + 1$, and if $v_i = 0$, then $x_{ri} \leq d_i - 1$. To see this, note that any $x_{ri} = 0$ will be preceded by a sequence $(d_i+1)d_i^l$, and blocks of the form $(d_i+1)d_i^l$ can be separated by blocks of the form 01^l, thus giving a special configuration.

Such a special configuration of v_i leads in $2k$ steps to a sequence of the form 01^l by Lemma 13, thus reducing each $x_{ri} \geq d_i + 1$ by at least $d_i - 1$.

Performing this transformation $O(V)$ times will ensure that all resulting x_{ri} have value at most d_i, and we thus have a final configuration in $O(kV)$ steps. □

Combining Lemmas 6, 14, 15, and 16, we obtain the following.

Theorem 5. *If $2 \leq d_i \leq d_j$ for $1 \leq i \leq j$, then the number of steps is bounded by $O(k\log_{d_{\min}}(kd_{\max}))$.*

Theorem 6. *If $d_i \geq d_j$ for $1 \leq i \leq j$, then the number of steps is at most $2k^2$.*

Proof. We may assume $d_0 = d_{\max}$ by Lemma 6. We wish to reach the configuration $x_i = e_i$. Suppose more generally we wish to reach a configuration $x_0 = e_0 + (d_0+1)u$, $x_i = e_i + (d_i - 1)u$ for $1 \leq i \leq k$. This configuration can be reached from $x'_0 = e_0 + (d_0+1)(u+1+(d_k-1)u)$, $x'_i = e_i + (d_i-1)(u+1+(d_k-1)u)$ for $1 \leq i \leq k-1$, $x'_k = e_k - 1$ in $2k-1$ steps by Lemma 13 that transforms $(d_0+1)d_1 \cdots d_{k-1}$ into 01^k. The configuration x'_i can in turn be reached from $x'_0 = e_0 + (d_0+1)(u+e_k+(d_k-1)u)$, $x'_i = e_i + (d_i-1)(u+e_k+(d_k-1)u)$ for $1 \leq i \leq k-1$, $x'_k = 0$ in $2k-1$ steps by Lemma 13 again.

We have thus obtained a configuration $x'_0 = e_0 + (d_0+1)v$, $x_i = e_i + (d_i - 1)v$ for $1 \leq i \leq k-1$ in $2(2k-1)$ steps, and this configuration has $x'_k = 0$. Repeatedly applying the same argument, we may set $x''_{k-1} = 0$, $x'''_{k-2} = 0, \ldots$ in turn, until the initial configuration is reached. The number of steps is $2(2k-1) + 2(2(k-1)-1) + 2(2(k-3)-1) + \cdots = 2k^2$. □

THEOREM 7. *In the general case of n arbitrary and all $d_i \geq 2$ for $i \geq 1$, the number of steps is bounded by $O(k^{1.5} + \log_{d_{\min}} n) = O(k^{1.5} + k \log_{d_{\min}} d_{\max})$.*

Proof. Consider the slowed-down fractional game of Lemma 11 that reaches depth k within $O(\log_{d_{\min}} n) = O(k \log_{d_{\min}} d_{\max})$ steps. At the end of this fractional game, we have $t_{ri} \leq d_i + s_{ri}$ as before, and the excesses s_{ri} account for strictly less than one disk as before. The d_i for $i < j$ together with the extra disk coming from the s_{ri} account for less than one disk that is in the subtree rooted at a vertex at depth j in the final solution. The reason is that a sequence $(d_0 + 1)d_1 d_2 \cdots d_l$ can at most transform to 01^{l+1}, with one disk moving to a subtree rooted at depth $l + 1$.

In the extreme case, suppose each subtree is missing one disk from its parent. If a leaf at depth k is going to receive 1 disk from its parent, then the parent at depth $k - 1$ must also send 1 disk to its parent and thus receive a total of 2 disks from its parent; then the parent at depth $k - 2$ must also send 2 disks to its parent and thus receive a total of 3 disks from its parent. In general, a vertex at depth $k - i$ will send at most i disks to its parent and receive at most $i + 1$ disks from its parent.

Consequently, in the combinatorial game, after the initial

$$O(\log_{d_{\min}} n) = O(k \log_{d_{\min}} d_{\max})$$

steps follows a second phase during which each vertex sends at most k disks to each of its neighbors, for a total of k^2 disks when adding over all depths i. As long as a vertex at some depth i has at least $\sqrt{k}(d_i + 1)$ disks, such a vertex will send $\sqrt{k}$ disks simultaneously to each of its neighbors; this can happen for at most $k^2/\sqrt{k} = k^{1.5}$ steps.

Once the vertices at each depth i have at most $\sqrt{k}(d_i + 1)$ disks, a final configuration can be reached in $O(k^{1.5})$ steps by Lemma 16, completing the proof. □

THEOREM 8. *Assume that $d_k = e_k$. If $d_i \geq 2$ for $i \geq 1$, then there is a lower bound of $\Omega(k + \max_{1 \leq i \leq k} \sum_{i \leq j \leq k} \log_{d_i} d_j)$ on the number of steps. Thus the upper bound of Theorem 4 is tight up to constant factors, provided $\log_{d_{\min}} d_1 = O(1)$, that is, $d_1 \leq d_{\min}^{O(1)}$ (since in that case $O(\sum_{i \leq k} \log_{d_{\min}} d_i) = O(\sum_{i \leq k} \log_{d_1} d_i)$).*

Proof. There is an immediate lower bound of $\Omega(k)$ to reach depth k. Consider the n_i disks that may reach depth i. Only a fraction $d_i/(d_i + 1)$ of these disks will be moved from a vertex at depth i to depth $i + 1$ at one time, since a fraction $1/(d_i + 1)$ must be moved to depth $i - 1$. Thus after one step, we are still left with at least $(n_i - e_i)/(d_i + 1)$ disks that have not yet reached depth $i + 1$; after two steps we are still left with at least $(n_i - e_i)/(d_i + 1)^2$ disks that have not yet reached depth $i + 1$; and after r steps we are still left with at least $(n_i - e_i)/(d_i + 1)^r$ disks that have not yet reached depth $i + 1$. It will thus take at least $r = \log_{d_i+1}(n_i - e_i)$ steps to move the $n_i - e_i$ disks to depth $i + 1$. The result follows from the bound $n_i \geq d_i d_{i+1} \cdots d_k$. □

5. Conclusion. We have analyzed a combinatorial game played on an infinite rooted tree where all the vertices at depth i have the same number of children d_i. The analysis determines the final configuration in the general case and bounds the number of steps needed to reach this final configuration when $d_i \geq 2$ for $i \geq 1$. The case where all $d_i = 1$ for $i \geq 1$ was previously studied by Anderson et al. [1]. It remains open to analyze the combinatorial game when some of the d_i for $i \geq 1$ satisfy $d_i = 1$ and some satisfy $d_i \geq 2$.

The fact that the dependence of the number of steps depends essentially linearly on the depth of the tree and logarithmically in the number of disks, instead of being quadratic as in the case of a path, indicates that the particular structure of each graph under consideration greatly affects the number of steps that the game takes. It thus seems that trees are the graphs for which the convergence to the final configuration is fastest, as it is in many cases only linear in the diameter reached by the game.

REFERENCES

[1] R. Anderson, L. Lovász, P. Shor, J. Spencer, E. Tardos, and S. Winograd, *Disks, balls, and walls: Analysis of a combinatorial game*, Amer. Math. Monthly, 96 (1989), pp. 481–493.

[2] N. L. Biggs, *Chip-firing and the critical group of a graph*, J. Algebraic Combin., 9 (1999), pp. 25–45.

[3] A. Björner, L. Lovász, and P. Shor, *Chip firing games on graphs*, European J. Combin., 12 (1991), pp. 283–291.

[4] A. Gabrielov, *Avalanches, sandpiles and Tutte decomposition*, in The Gelfand Mathematical Seminars, 1990-1992, Birkhäuser, Boston, 1993, pp. 19–26.

[5] D. Dhar, *Self-organized critical state of sandpile automaton models*, Phys. Rev. Lett., 64 (1990) pp. 1613–1616.

[6] D. Dhar and S. N. Majumdar, *Abelian sandpile model on the Bethe lattice*, J. Phys. A, 23 (1990), pp. 4333–4350.

[7] D. Dhar and S. N. Majumdar, *Height correlations in the Abelian sandpile model*, J. Phys. A, 24 (1991), pp. L357–L362.

[8] D. Dhar, *The Abelian sandpile model of self-organized criticality*, in Computer-Aided Studies of Statistical Physics, AIP Conf. Proc. 248, C. K. Hu, ed., AIP, New York, 1992, pp. 226–231.

[9] D. Dhar and S. N. Majumdar, *Equivalence of the sandpile model and the $q \to 0$ limit of the Potts model*, Phys. A, 185 (1992), pp. 129–145.

[10] D. Dhar and S. S. Manna, *Inverse avalanches in the Abelian sandpile model*, Phys. Rev. E, 49 (1994), pp. 2684–2687.

[11] A. A. Ali and D. Dhar, *Breakdown of the simple scaling in the Abelian sandpile models in one dimension*, Phys. Rev. E, 51 (1995), pp. R2705–R2708.

[12] D. Dhar, P. Ruelle, S. Sen, and D. N. Verma, *Algebraic aspects of Abelian sandpile models*, J. Phys. A, 28 (1995), pp. 805–831.

[13] A. A. Ali and D. Dhar, *Structure of avalanches and breakdown of simple scaling in Abelian sandpile model in one dimension*, Phys. Rev. E, 52 (1995), pp. 4804–4816.

[14] D. Dhar, *Extended operator algebra for Abelian sandpile models*, Phys. A, 224 (1996), pp. 162–168.

[15] D. Dhar and B. Tadic, *Emergent spatial structures in critical sandpiles*, Phys. Rev. Lett., 79 (1997), pp. 1519–1522.

[16] D. Dhar, *The Abelian sandpile and related models*, Phys. A, 263 (1999), p. 4.

[17] D. Dhar, *Some results and a conjecture for Manna's stochastic sandpile model*, Phys. A, 270 (1999), p. 69.

[18] D. Dhar and S. Lubeck, *Continuously varying critical exponents in a sandpile model with dissipation near surface*, J. Statist. Phys., 102 (2001), pp. 1–14.

[19] P. K. Mohanty and D. Dhar, *Generic sandpile models have directed percolation exponents*, Phys. Rev. Lett., 89 (2002), 104303.

[20] E. Goles, M. Morvan, and H. D. Phan, *The structure of a linear chip firing game and related models*, Theoret. Comput. Sci., 270 (2002), pp. 827–841.

[21] E. Goles and E. Prisner, *Source reversal and chip firing on graphs*, Theoret. Comput. Sci., 233 (2000), pp. 287–295.

[22] E. Goles and M. Margenstern, *Universality of the chip-firing game*, Theoret. Comput. Sci., 172 (1997), pp. 121–134.

[23] M. A. Kiwi, R. Ndoundam, M. Tchuente, and E. Goles, *No polynomial bound for the period of the parallel chip firing game on graphs*, Theoret. Comput. Sci., 136 (1994), pp. 527–532.

[24] E. Goles and M. A. Kiwi, *Games on line graphs and sand piles*, Theoret. Comput. Sci., 115 (1993), pp. 321–349.

[25] E. Goles and M. A. Kiwi, *Dynamics of sand-piles games on graphs*, in LATIN '92, Lecture Notes in Comput. Sci. 583, I. Simon, ed., Springer-Verlag, Berlin, 1992, pp. 219–230.

[26] J. Bitar and E. Goles, *Parallel chip firing games on graphs*, Theoret. Comput. Sci., 92 (1992), pp. 291–300.

[27] G. Tardos, *Polynomial bound for a chip firing game on graphs*, SIAM J. Discrete Math., 1 (1988), pp. 397–398.

SIAM J. Discrete Math.
Vol. 19, No. 3, pp. 553–574

ALGORITHMS FOR PERFECTLY CONTRACTILE GRAPHS[*]

FRÉDÉRIC MAFFRAY[†] AND NICOLAS TROTIGNON[†]

Abstract. We consider the class $\mathcal{A}$ of graphs that contain no odd hole, no antihole of length at least 5, and no prism (a graph consisting of two disjoint triangles with three disjoint paths between them) and the class $\mathcal{A}'$ of graphs that contain no odd hole, no antihole of length at least 5, and no odd prism (prism whose three paths are odd). These two classes were introduced by Everett and Reed and are relevant to the study of perfect graphs. We give polynomial-time recognition algorithms for these two classes. In contrast we prove that determining if a general graph contains a prism (or an even prism, or an odd prism) is NP-complete.

Key words. perfect graph, even pair, perfectly contractile, recognition algorithm

AMS subject classifications. 05C17, 05C85

DOI. 10.1137/S0895480104442522

1. Introduction.

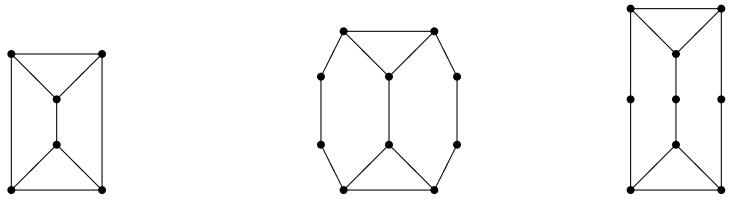
A graph G is *perfect* if every induced subgraph G' of G satisfies $\chi(G') = \omega(G')$, where $\chi(G')$ is the chromatic number of G' and $\omega(G')$ is the maximum clique size in G'. Berge [1, 2, 3] introduced perfect graphs and conjectured that *a graph is perfect if and only if it does not contain as an induced subgraph an odd hole or an odd antihole* (the strong perfect graph conjecture), where a *hole* is a chordless cycle with at least four vertices and an *antihole* is the complement of a hole. We follow the tradition of calling a *Berge graph* any graph that contains no odd hole and no odd antihole. The strong perfect graph conjecture was the object of much research (see [13]) until it was finally proved by Chudnovsky et al. [7]: *every Berge graph is perfect.* Moreover, Chudnovsky et al. [6] gave polynomial-time algorithms to decide if a graph is Berge.

Despite those breakthroughs, some conjectures about Berge graphs remain open. An *even pair* in a graph G is a pair of nonadjacent vertices such that every chordless path between them has even length (number of edges). Given two vertices x, y in a graph G, the operation of *contracting* them means removing x and y and adding one vertex with edges to every vertex of $G \setminus \{x, y\}$ that is adjacent in G to at least one of x, y; we denote by G/xy the graph that results from this operation. Fonlupt and Uhry [9] proved that *if G is a perfect graph and $\{x, y\}$ is an even pair in G, then the graph G/xy is perfect and has the same chromatic number as G.* In particular, given a $\chi(G/xy)$-coloring c of the vertices of G/xy, one can easily obtain a $\chi(G)$-coloring of the vertices of G as follows: keep the color for every vertex different from x, y; assign to x and y the color assigned by c to the contracted vertex. This idea could be the basis for a conceptually simple coloring algorithm for Berge graphs: as long as the graph has an even pair, contract any such pair; when there is no even pair, find a coloring c of the contracted graph and, applying the procedure above repeatedly, derive from c a coloring of the original graph. The polynomial-time algorithm for recognizing Berge graphs mentioned at the end of the preceding paragraph can be used to detect an even pair in a Berge graph G; indeed, two nonadjacent vertices a, b

[*] Received by the editors March 29, 2004; accepted for publication (in revised form) January 4, 2005; published electronically November 15, 2005.

http://www.siam.org/journals/sidma/19-3/44252.html

[†] C.N.R.S., Laboratoire Leibniz-IMAG, 46 Avenue Félix Viallet, 38031 Grenoble Cedex, France (frederic.maffray@imag.fr, nicolas.trotignon@imag.fr). The work of the second author was supported by Université Pierre Mendes France, Grenoble.



An odd prism An odd prism An even prism

FIG. 1. *Some prisms.*

form an even pair in G if and only if the graph obtained by adding a vertex adjacent only to a and b is Berge. The problem of deciding if a graph contains an even pair is NP-hard in general graphs [5]. Given a Berge graph G, one can try to color its vertices by contracting even pairs until none can be found. Then some questions arise: what are the Berge graphs with no even pair? what are, on the contrary, the graphs for which a sequence of even-pair contractions leads to graphs that are easy to color?

As a first step toward getting a better grasp on these questions, Bertschi [4] proposed the following definitions. A graph G is *even-contractile* if either G is a clique or there exists a sequence $G_0, \ldots, G_k$ of graphs such that $G = G_0$, for $i = 0, \ldots, k-1$ the graph G_i has an even pair $\{x_i, y_i\}$ such that $G_{i+1} = G_i/x_iy_i$, and G_k is a clique. A graph G is *perfectly contractile* if every induced subgraph of G is even-contractile. Perfectly contractile graphs include many classical families of perfect graphs, such as Meyniel graphs, weakly chordal graphs, and perfectly orderable graphs; see [8]. Everett and Reed proposed a conjecture aiming at a characterization of perfectly contractile graphs. To understand it, one more definition is needed: say that a graph is a *prism* if it consists of two vertex-disjoint triangles (cliques of size 3) $\{a_1, a_2, a_3\}$, $\{b_1, b_2, b_3\}$, with three vertex-disjoint paths P_1, P_2, P_3 between them, such that for $i = 1, 2, 3$ path P_i is from a_i to b_i, and with no other edge than those in the two triangles and in the three paths. We may also say that the three paths P_1, P_2, P_3 *form* the prism. Say that a prism is odd (or even) if all three paths have odd length (respectively, all have even length). See Figure 1.

Define two classes $\mathcal{A}$, $\mathcal{A}'$ of graphs as follows:

- $\mathcal{A}$ is the class of graphs that do not contain odd holes, antiholes of length at least 5, or prisms.
- $\mathcal{A}'$ is the class of graphs that do not contain odd holes, antiholes of length at least 5, or odd prisms.

Clearly $\mathcal{A} \subset \mathcal{A}'$. Class $\mathcal{A}$ was called Artemis graphs in [8, 14].

CONJECTURE 1 (see [8, 14]). *A graph is perfectly contractile if and only if it is in class $\mathcal{A}'$.*

The *if* part of this conjecture remains open. The *only if* part is not hard to establish, but it requires some careful checking; this was done formally in [11]. A weaker form of this conjecture was also proposed by Everett and Reed; that statement is now a theorem.

THEOREM 1.1 (Maffray and Trotignon [12]). *If G is a graph in class $\mathcal{A}$ and G is not a clique, then G has an even pair whose contraction yields a graph in $\mathcal{A}$ (and so G is perfectly contractile).*

The preceding conjecture and theorem suggest that it may be interesting to recognize the classes $\mathcal{A}$ and $\mathcal{A}'$ in polynomial time; this is the aim of this paper.

To decide if a graph is in class $\mathcal{A}$, it would suffice to decide separately if it is Berge, if it has an antihole of length at least 5, and if it contains a prism. The first question, deciding if a graph is Berge, is now settled [6]. In section 2 we will find it convenient for our purpose to give a summary of the polynomial-time algorithm from [6] that solves this problem. The second question is not hard: to decide if a graph G contains a hole of length at least 5, it suffices to test, for every chordless path a-b-c, whether a and c are in the same connected component of the subgraph of G obtained by removing the vertices of $N(a) \cap N(c)$ and those of $N(b) \setminus \{a, c\}$. This takes time $O(|V(G)|^5)$. To decide if a graph contains an antihole of length at least 5, we need only apply this algorithm on its complementary graph. However, the third question, to decide if a graph contains a prism, turns out to be NP-complete; this is established in section 8 below. Likewise, we will see that it is NP-complete to decide if a graph contains an odd prism. Thus we cannot solve the recognition problem for class $\mathcal{A}$ (or for class $\mathcal{A}'$) in the fashion that is suggested at the beginning of this paragraph. Instead, we will adapt parts of the Berge graph recognition algorithm to our purpose. This is done in sections 3–7.

2. Recognizing Berge graphs.

We give here a brief outline of the Berge graph recognition algorithm from [6]. Given a graph G and a hole C in G, a vertex $x \in V(G) \setminus V(C)$ is called *C-major* if the set $N(x) \cap V(C)$ is not included in a 3-vertex subpath of C, and a set $X \subseteq V(G)$ is called a *near-cleaner* for C if X contains all the C-major vertices and $X \cap V(C)$ is included in a 3-vertex subpath of C. The algorithm is based on the results summarized in the following theorem.

THEOREM 2.1 (see [6]).

1. *There exist five types of configurations (graphs) such that, for $i = 1, \dots, 5$, we have the following:* (a) *if a graph G contains a configuration of type i, then G is not a Berge graph, and* (b) *there is a polynomial-time algorithm Ai that decides if a graph contains a configuration of type i.*

2. *There is a polynomial-time algorithm which, given a graph G that does not contain a configuration of any of the five types, returns a family $\mathcal{F}$ of $|V(G)|^5$ subsets of $V(G)$ such that for any shortest odd hole C of G, some member of $\mathcal{F}$ is a near-cleaner for C.*

3. *There is a polynomial-time algorithm which, given a graph G that does not contain a configuration of any of the five types and the family $\mathcal{F}$ produced by step* 2, *decides if G contains an odd hole (and if it does, returns a shortest odd hole of G).*

The five types of configurations are called types $\mathcal{T}_1$, $\mathcal{T}_2$, $\mathcal{T}_3$, *jewel*, and *pyramid*. We will not give the definition of all of them, but we recall that for $i = 1, \dots, 5$, the complexity of algorithm Ai given in [6] is, respectively, $O(|V(G)|^5)$, $O(|V(G)|^6)$, $O(|V(G)|^6)$, $O(|V(G)|^6)$, $O(|V(G)|^9)$. We need to dwell on the configuration that is called a *pyramid*. A pyramid is a graph that consists of three pairwise adjacent vertices b_1, b_2, b_3 (called the triangle vertices of the pyramid), a fourth vertex a (called the apex of the pyramid), and three chordless paths P_1, P_2, P_3 such that

- for $i = 1, 2, 3$, path P_i is between a and b_i;
- for $1 \leq i < j \leq 3$, $V(P_i) \cap V(P_j) = \{a\}$ and $b_i b_j$ is the only edge between $V(P_i) \setminus \{a\}$ and $V(P_j) \setminus \{a\}$;
- a is adjacent to at most one of b_1, b_2, b_3.

We may say that the three paths P_1, P_2, P_3 *form* a pyramid. It is easy to see that a pyramid contains an odd hole (since two of the paths P_1, P_2, P_3 have the same parity, the union of their vertex sets induces an odd hole); thus Berge graphs do not contain pyramids. The pyramid-testing algorithm from [6] is the slowest algorithm in step 1 of the Berge graph recognition algorithm. The algorithm of step 2 has complexity $O(|V(G)|^5)$ [6], and the algorithm of step 3 has complexity $O(|V(G)|^9)$ [6]. Testing if a graph G is Berge can be done by running the algorithms described in the previous theorem on G and on its complementary graph $\overline{G}$. Thus the total complexity is $O(|V(G)|^9)$.

3. Recognizing pyramids and prisms. We present a polynomial-time algorithm that decides if a graph contains a pyramid or a prism. This algorithm has the same flavor as the pyramid-testing algorithm from [6]. We describe this algorithm now.

If a graph contains a pyramid or a prism, it contains a pyramid or a prism that is *smallest* in the sense that there is no pyramid or prism induced by strictly fewer vertices. Smallest pyramids or prisms have properties that make them easier to handle. These properties are expressed in the next two lemmas.

Whenever we deal with a chordless path P in a graph G, and a, b are two vertices of P, we use a-P-b to denote the subpath of P whose endvertices are a, b.

LEMMA 3.1. *Let G be a graph. Let K be a smallest pyramid or prism in G. Suppose that K is a pyramid, formed by paths P_1, P_2, P_3, with triangle $\{b_1, b_2, b_3\}$ and apex a. Let R_1 be a shortest path from b_1 to a whose interior vertices are not adjacent to b_2 or b_3. Then the subgraph induced by $V(R_1) \cup V(P_2) \cup V(P_3)$ is a smallest pyramid or prism in G.*

Proof. Note that $|V(R_1)| \leq |V(P_1)|$ since P_1 is a path from b_1 to a whose interior vertices are not adjacent to b_2 or b_3. Let P be the path induced by $(V(P_2) \setminus \{b_2\}) \cup (V(P_3) \setminus \{b_3\})$. If no vertex of $R_1 \setminus \{a\}$ has any neighbor in $P \setminus \{a\}$, then R_1, P_2, P_3 form a pyramid in G, and its number of vertices is not larger than $|V(K)|$, so the lemma holds. So we may assume that some vertex c of $R_1 \setminus \{a\}$ has a neighbor in $P \setminus \{a\}$, and we choose c closest to b_1 along R_1. Note that this choice ensures that no vertex of the path b_1-R_1-c is in $P \setminus \{a\}$. Recall that c is not adjacent to b_2 or b_3, by the definition of R_1. For $j = 2, 3$, let b'_j be the neighbor of b_j along P_j (so b'_2, b'_3 are the ends of P) and let c_j be the neighbor of c closest to b'_j along P.

Suppose $c_2 = c_3$. We have $c_3 \neq a$ since c has a neighbor along $P \setminus \{a\}$. Then the three chordless paths c_2-c-R_1-b_1, c_2-P-b_2, c_2-P-b_3 form a pyramid with triangle $\{b_1, b_2, b_3\}$ and apex c_2; this pyramid is strictly smaller than K, because it is included in $(V(R_1) \setminus \{a\}) \cup V(P_2) \cup V(P_3)$, a contradiction. So $c_2 \neq c_3$. If c_2, c_3 are not adjacent, then the three chordless paths c-R_1-b_1, c-c_2-P-b_2, c-c_3-P-b_3 form a pyramid with triangle $\{b_1, b_2, b_3\}$ and apex c; again this pyramid has strictly fewer vertices than K, a contradiction. So c_2, c_3 are adjacent. Then the three chordless paths c-R_1-b_1, c_2-P-b_2, and c_3-P-b_3 form a prism K', with triangles $\{b_1, b_2, b_3\}$ and $\{c, c_2, c_3\}$. If $a \notin \{c_2, c_3\}$, then K' is smaller than K, a contradiction. So $a \in \{c_2, c_3\}$ and the prism K' has the same size as K, so the lemma holds. □

LEMMA 3.2. *Let G be a graph. Let K be a smallest pyramid or prism in G. Suppose that K is a prism, formed by paths P_1, P_2, P_3, with triangles $\{a_1, a_2, a_3\}$ and $\{b_1, b_2, b_3\}$, so that, for $i = 1, 2, 3$, path P_i is from a_i to b_i. Then*

1. *If R_1 is any shortest path from a_1 to b_1 whose interior vertices are not adjacent to b_2 or b_3, then R_1, P_2, P_3 form a prism of size $|V(K)|$ in G, with triangles $\{a_1, a_2, a_3\}$ and $\{b_1, b_2, b_3\}$.*
2. *If R_2 is any shortest path from a_1 to b_2 whose interior vertices are not adjacent to b_1 or b_3, then either the three paths $P_1, R_2 \setminus a_1, P_3$ form a smallest prism in G, or the three paths $P_1, R_2, P_3 + a_1$ form a pyramid of size $|V(K)|$ in G, with triangle $\{b_1, b_2, b_3\}$ and apex a_1.*

Proof. Let us prove the first item of the lemma. Note that $|V(R_1)| \leq |V(P_1)|$ since P_1 is a path from a_1 to b_1 whose interior vertices are not adjacent to b_2 or b_3. Let P be the path induced by $(V(P_2) \setminus \{b_2\}) \cup (V(P_3) \setminus \{b_3\})$. If no interior vertex of R_1 is adjacent to any vertex of $V(P)$, then the three paths R_1, P_2, P_3 form a prism in G whose size is not larger than the size of K, so it must be a smallest prism and the lemma holds. So we may assume that there is an interior vertex c of R_1 that has a neighbor in $V(P)$ and we choose c closest to b_1 along R_1. For $j = 2, 3$, let b'_j be the neighbor of b_j along P_j (so b'_2, b'_3 are the ends of P) and let c_j be the neighbor of c closest to b'_j along P.

Suppose $c_2 = c_3$. Then the three paths c_2-c-R_1-b_1, c_2-P-b_2, c_2-P-b_3 form a pyramid with triangle $\{b_1, b_2, b_3\}$ and apex c_2; this pyramid is strictly smaller than K (since $|V(R_1 \setminus \{a\})| < |V(P_1)|$), a contradiction. Thus, $c_2 \neq c_3$. If c_2, c_3 are adjacent, then the three paths c-R_1-b_1, c_2-P-b_2, c_3-P-b_3 form a prism, with triangles $\{b_1, b_2, b_3\}$ and $\{c, c_2, c_3\}$, that is strictly smaller than K, a contradiction. Thus, c_2, c_3 are not adjacent. But then the three paths c-R_1-b_1, c-c_2-P-b_2, c-c_3-P-b_3 form a pyramid with triangle $\{b_1, b_2, b_3\}$, apex c, and this pyramid is strictly smaller than K, a contradiction. Therefore, the first item is proved.

Now we prove the second item of the lemma. Note that $|V(R_2)| \leq |V(P_2)| + 1$ since $P_2 + a_1$ is a path from a_1 to b_2 whose interior vertices are not adjacent to b_2 or b_3. Let P be the path induced by $(V(P_1) \setminus \{b_1\}) \cup (V(P_3) \setminus \{b_3\})$. If no interior vertex of R_2 has any neighbor in $V(P \setminus a_1)$, then $P_1, R_2, P_3 + a_1$ form a pyramid, which is not larger than K; so it is a smallest pyramid and the theorem holds. Now assume that some interior vertex of R_2 has a neighbor in $V(P)$, and choose the vertex c that has this property and is closest to b_2. For $i = 1, 3$, let b'_i be the neighbor of b_i along P_i (so b'_1, b'_3 are the ends of P) and let c_i be the neighbor of c along P that is closest to b'_i.

Suppose $c_1 = c_3$. Then $c_1 \neq a_1$ since c has a neighbor in $V(P \setminus a_1)$. Then the three paths c_1-c-R_2-b_2, c_1-P-b_1, c_1-P-b_3 from a pyramid with triangle $\{b_1, b_2, b_3\}$ and apex c_1. This pyramid is strictly smaller than K, a contradiction. Thus, $c_1 \neq c_3$. If c_1, c_3 are not adjacent, then the three paths c-R_2-b_2, c-c_1-P-b_1, c-c_3-P-b_3 form a pyramid with triangle $\{b_1, b_2, b_3\}$ and apex c; this pyramid has size strictly smaller than K, a contradiction. So c_1, c_3 are adjacent. Then the three paths c-R_2-b_2, c_1-P-b_1, c_3-P-b_3 form a prism K', with triangles $\{b_1, b_2, b_3\}$ and $\{c, c_1, c_3\}$. If $a_1 \notin \{c_1, c_3\}$, then this prism is strictly smaller than K, a contradiction. So $a_1 \in \{c_1, c_3\}$ and K' has the same size as K, and the lemma holds. This completes the proof of the lemma. □

On the basis of the preceding lemmas we can present an algorithm for testing if a graph contains a pyramid or a prism.

Algorithm 1. *Detection of a pyramid or prism.*

Input: *A graph G.*

Output: *An induced pyramid or prism of G, if G contains any; else the negative answer "G contains no pyramid and no prism."*

Method: *For every quadruple a, b_1, b_2, b_3 of vertices of G such that b_1, b_2, b_3 are pairwise adjacent and a is adjacent to at most one of them, do: Compute a shortest path P_1 from a to b_1 whose interior vertices are not adjacent to b_2, b_3, if any. Compute paths P_2 and P_3 similarly. If the three paths P_1, P_2, P_3 exist, and if $V(P_1) \cup V(P_2) \cup V(P_3)$ induces a pyramid or a prism, then return this subgraph of G, and stop.*

If no quadruple has produced a pyramid or a prism, return the negative answer.

Complexity: $O(|V(G)|^6)$.

Proof of correctness. If G contains no pyramid and no prism then clearly the algorithm will return the negative answer. Conversely, suppose that G contains a pyramid or a prism. Let K be a smallest pyramid or prism. Let b_1, b_2, b_3 be the vertices of a triangle of K, and let a be such that if K is a pyramid, then a is its apex, and if K is a prism, then a is a vertex of the other triangle of K. When our algorithm considers the quadruple a, b_1, b_2, b_3, it will find paths P_1, P_2, P_3 since some paths in K do have the required properties. Then, three applications of Lemmas 3.1 and 3.2 imply that P_1, P_2, P_3 do form a pyramid or a prism of G. So the algorithm will detect this subgraph.

Complexity analysis. Testing all quadruples takes time $O(|V(G)|^4)$. For each quadruple, finding the three paths takes time $O(|V(G)|^2)$ and checking that the corresponding subgraph is a pyramid or prism takes time $O(|V(G)|^2)$. Thus the overall complexity is $O(|V(G)|^6)$.

We now show how the results of the preceding algorithm can be performed a little faster.

LEMMA 3.3. *Let G be a graph, and let $\{a, b, c\}$ be a triangle in G. Suppose that $G \setminus \{a, b, c\}$ is connected, that each of a, b, c has exactly one neighbor a', b', c' in $G \setminus \{a, b, c\}$, and that a', b', c' are pairwise distinct. Then G contains a prism or a pyramid with triangle $\{a, b, c\}$.*

Proof. Let P be a shortest path in $G \setminus \{a, b, c\}$ from a' to b'. Let Q be a path in $G \setminus \{a, b, c\}$ from c' to a vertex w that has a neighbor in P, and choose such a Q of minimal length. Clearly P and Q exist since $G \setminus \{a, b, c\}$ is connected. Note that a', b', and w are distinct. (Possibly $c' \in V(P)$, and in that case $w = c'$. If $c' \notin V(P)$, then $w \notin V(P)$.) Let x be the neighbor of w along P that lies closest to a', and y be the neighbor of w along P that lies closest to b'. If $x = y$, then the three paths x-w-Q-c'-c, x-P-a'-a, and x-P-b'-b form a pyramid with apex x and triangle $\{a, b, c\}$. If x, y are distinct and adjacent, then the three paths w-Q-c'-c, x-P-a'-a, and y-P-b'-b form a prism with triangles $\{a, b, c\}$ and $\{w, x, y\}$. If x, y are distinct and not adjacent, then the three paths w-Q-c'-c, w-x-P-a'-a, and w-y-P-b'-b form a pyramid with apex w and triangle $\{a, b, c\}$. □

Now we can give an algorithm.

ALGORITHM 2. *Detection of a pyramid or prism.*

Input: *A graph G.*

Output: *The positive answer "G contains a pyramid or a prism" if it does; else the negative answer "G contains no pyramid and no prism."*

Method: *For every triple b_1, b_2, b_3 of pairwise adjacent vertices of G do:*

Step 1. *Compute the set X_1 of those vertices of $V(G)$ that are adjacent to b_1 and not adjacent to b_2 or b_3, and the similar sets X_2, X_3, and compute the set X of those vertices of $V(G)$ that are not adjacent to any of b_1, b_2, b_3. Compute the connected components of X in G. For each component H of X, and for $i = 1, 2, 3$, if some vertex of H has a neighbor in X_i, then mark H with label i.*

Step 2. *For every component H of X that has received label $i \in \{1, 2, 3\}$, and for every vertex x of X_i that has a neighbor in H, assign to x the other labels of H (if any). For each $i = 1, 2, 3$ and for every vertex x of X_i that has a neighbor in X_j with $j \in \{1, 2, 3\}$ and $j \neq i$, assign label j to x.*

Step 3. *If some vertex of $X_1 \cup X_2 \cup X_3$ gets two different labels, return the positive answer and stop.*

If the positive answer has not been returned at Step 3, *return the negative answer.*

Complexity: $O(|V(G)|^5)$.

Proof of correctness. Suppose that G contains a pyramid or a prism K. Let

b_1, b_2, b_3 be the vertices of a triangle of K, and for $i = 1, 2, 3$ let c_i be the neighbor of b_i in $K \setminus \{b_1, b_2, b_3\}$. Let us observe what the algorithm will do when it examines the triple $\{b_1, b_2, b_3\}$. The algorithm will place the three vertices c_1, c_2, c_3 in the sets X_1, X_2, X_3, respectively. We claim that c_1 receives label 2 at Step 2. Indeed, if c_1 is adjacent to c_2 this is clear. Else, consider the (unique) path R from c_1 to c_2 in $K \setminus \{b_1, b_2\}$. The interior vertices of R lie in one component H of X, which will therefore get labels 1 and 2 at Step 1 (because of c_1, c_2), and so c_1 will get label 2 at Step 2. This proves the claim. Similarly, c_1 will get label 3. So the algorithm will return the positive answer.

Conversely, suppose that the algorithm returns the positive answer when it is examining a triple $\{b_1, b_2, b_3\}$ that induces a triangle of G. So (up to symmetry) some vertex $c_1 \in X_1$ gets labels 2 and 3 at Step 2. This means that for $j = 2, 3$, there exists a path R_j from c_1 to a vertex of X_j such that the interior vertices of R_j (if any) lie in X. We can apply Lemma 3.3 to the subgraph induced by $V(R_2) \cup V(R_3) \cup \{b_1, b_2, b_3\}$ with respect to the triangle $\{b_1, b_2, b_3\}$, which implies that this subgraph (and thus G itself) contains a pyramid or a prism. This completes the proof of correctness.

Complexity analysis. Finding all triples takes time $O(|V(G)|^3)$. For each triple, computing the sets X_1, X_2, X_3, X takes time $O(|V(G)|)$. Finding the components of X takes time $O(|V(G)|^2)$. Marking the components at the end of Step 1 can be done as follows. For each edge uv of G, if u is in a component H of X and v is in some X_i then mark H with label i. This takes time $O(|V(G)|^2)$. Marking the vertices of $X_1 \cup X_2 \cup X_3$ at Step 2 can be done similarly. Thus the overall complexity is $O(|V(G)|^5)$.

We observe that the above two algorithms are faster than the algorithm from [6] for finding a pyramid.

4. Recognition of graphs in class $\mathcal{A}$. We can now present the algorithm for recognizing graphs in the class $\mathcal{A}$.

ALGORITHM 3. *Recognition of graphs in class $\mathcal{A}$.*

Input: *A graph G.*

Output: *The positive answer "G is in class $\mathcal{A}$" if it is; else the negative answer "G is not in class $\mathcal{A}$."*

Method:

Step 1. *Test whether G contains no antihole of length at least* 5 *as explained at the end of the introduction.*

Step 2. *Test whether G has no pyramid or prism using Algorithm* 2 *above.*

Step 3. *Test whether G is Berge using the algorithm from the preceding section.*

Complexity: $O(|V(G)|^9)$.

The correctness of the algorithm is immediate from the correctness of the algorithms it refers to and from the fact that Berge graphs contain no pyramid. The complexity is dominated by the last step of the Berge recognition algorithm, which is $O(|V(G)|^9)$. Note that the other step of complexity $O(|V(G)|^9)$ in the Berge recognition algorithm (deciding if the input graph contains a pyramid) can be replaced by Step 2. Additionally, we can remark that it is not necessary to test for the existence of configurations of types T1, ..., T4 when we call the Berge recognition algorithm, because—this is not very hard to prove—any such configuration contains an antihole of length at least 5, so it is already excluded by Step 2. But this does not bring the overall complexity down from $O(|V(G)|^9)$.

The algorithm for recognizing graphs in class $\mathcal{A}$ can also be used to color graphs in class $\mathcal{A}$. Recall that Theorem 1.1 states that *if a graph G is in class $\mathcal{A}$ and is* *not a clique, it admits a pair of vertices whose contraction yields a graph in class $\mathcal{A}$.* Therefore we could enumerate all pairs of nonadjacent vertices of G and test whether their contraction produces a graph in class $\mathcal{A}$; Theorem 1.1 ensures that at least one pair will work. We can then iterate this procedure until the contractions turn the graph into a clique. Since each vertex of the clique is the result of contracting a stable set of G, a coloring of this clique corresponds to an optimal coloring of G. In terms of complexity, we may need to check $O(|V(G)|^2)$ pairs at each contraction step, and there may be $O(|V(G)|)$ steps. So we end up with complexity $O(|V(G)|^{12})$. This is not as good as the direct method from [12], whose complexity is $O(|V(G)|^6)$. (In fact the complexity of that method can be brought down to $O(|V(G)|^2|E(G)|)$, as pointed out by Bruce Reed to the authors; see [15].)

5. Even prisms. In this section we show how to decide in polynomial-time if a graph that contains no odd hole contains an even prism. Let K be an even prism, formed by paths P_1, P_2, P_3 with triangles $\{a_1, a_2, a_3\}$ and $\{b_1, b_2, b_3\}$ so that for $1 \le i \le 3$ path P_i is from a_i to b_i. Let m_i be the middle vertex of path P_i. We say that the 9-tuple $(a_1, a_2, a_3, b_1, b_2, b_3, m_1, m_2, m_3)$ is the *frame* of K. When we talk about a prism, the word small refers to its number of vertices.

LEMMA 5.1. *Let G be a graph that contains no odd hole and contains an even prism, and let K be a smallest even prism in G. Let K be formed by paths P_1, P_2, P_3 and have frame $(a_1, a_2, a_3, b_1, b_2, b_3, m_1, m_2, m_3)$ with $a_i, m_i, b_i \in V(P_i)$ $(1 \le i \le 3)$. Let R be any path of G whose ends are a_1, m_1 whose interior vertices are not adjacent to $a_2, a_3, b_2,$ or b_3 and which is shortest with these properties. Then a_1-R-m_1-P_1-b_1 is a chordless path R_1 and R_1, P_2, P_3 form a smallest even prism in G.*

Proof. Let k be the length (number of edges) of path P_1; so k is even. Note that $|E(R)| \le k/2$ since the path a_1-P_1-m_1 satisfies the properties required for R. Call Q the chordless path induced by $V(P_2) \cup V(P_3) \setminus \{a_2, a_3\}$ and call a'_2, a'_3 the ends of Q so that for $j = 2, 3$ vertex a'_j is adjacent to a_j.

Suppose that no interior vertex of R has any neighbor in Q. Let R' be a shortest path from a_1 to b_1 contained in a_1-R-m_1-P_1-b_1. So $|E(R')| \le k$ and R', P_2, P_3 form a prism K' with $|V(K')| \le |V(K)|$. Since G contains no odd hole, R' has even length (else $V(R') \cup V(P_2)$ would induce an odd hole), so K' is an even prism. Thus K' is a smallest even prism, and we have equality in the above inequalities; in particular, R' is equal to a_1-R-m_1-P_1-b_1 and the theorem holds.

We may now assume that some vertex c of R has a neighbor in Q, and we choose c closest to m_1 along R. Let S be a chordless path from c to b_1 contained in c-R-m_1-P_1-b_1. We have $|E(S)| < k$ since $|E(R)| \le k/2$ and $c \ne a_1$. By the choice of c, no vertex of $S \setminus b_1$ has a neighbor in P_2 or P_3. Let x, y be the neighbors of c along Q that are closest, respectively, to a'_2 and to a'_3. If $x = y$, then $V(S) \cup V(P_2) \cup V(P_3)$ induces a pyramid with triangle $\{b_1, b_2, b_3\}$ and apex x, so G contains an odd hole, a contradiction. Thus $x \ne y$. If x, y are not adjacent, then $V(S) \cup V(P_2) \cup V(P_3)$ contains a pyramid with triangle $\{b_1, b_2, b_3\}$ and apex c, a contradiction. So x, y are different and adjacent and, up to symmetry and since c is not adjacent to b_2, b_3, we may assume that x, y lie in the interior of P_2. Now $V(S) \cup V(P_2) \cup V(P_3)$ induces a prism K', with triangles $\{b_1, b_2, b_3\}$ and $\{c, x, y\}$, and $|V(K')| < |V(K)|$ since $|E(S)| < k$. Thus K' is an odd prism, which means that y-P_2-b_2 is an odd path, and so a_2-P_2-x is an even path. Let R'' be a chordless path from c to a_1 contained in c-R-m_1-P_1-a_1. We have $|E(R'')| < k$ since $|E(R)| \le k/2$ and $c \ne a_1$. By the choice of c no vertex of $R'' \setminus a_1$ has a neighbor in P_2 or P_3. Then R'' has even length for otherwise $V(R'') \cup V(a_2\text{-}P_2\text{-}x)$ induces an odd hole. Now $V(R'') \cup V(P_2) \cup V(P_3)$ induces a prism K'' with triangles

$\{a_1, a_2, a_3\}$ and $\{c, x, y\}$, and K'' is an even prism, and we have $|V(K'')| < |V(K)|$ since $|E(R'')| < k$. This is a contradiction, which completes the proof. □

Now we can give an algorithm.

ALGORITHM 4. *Detection of an even prism in a graph that contains no odd hole.*

Input: *A graph G that contains no odd hole.*

Output: *An induced even prism of G if G contains any; else the negative answer "G does not contain an even prism."*

Method: *For every 6-tuple $(a_1, a_2, a_3, b_1, b_2, b_3)$ of vertices of G such that the sets $\{a_1, a_2, a_3\}$ and $\{b_1, b_2, b_3\}$ induce disjoint triangles with no edge between them, do:*
Step 1. For $i = 1, 2, 3$, compute the set F_i of those vertices that are not adjacent to $a_{i+1}, a_{i+2}, b_{i+1}, b_{i+2}$ (with indices modulo 3); for each $m \in V(G)\setminus\{a_1, a_2, a_3, b_1, b_2, b_3\}$ look for a shortest path $R_i(m)$ from a_i to m whose interior vertices are in F_i, and look for a shortest path $S_i(m)$ from m to b_i whose interior vertices are in F_i. If $R_i(m)$ and $S_i(m)$ exist and their union is a chordless path from a_i to b_i, then call this path $P_i(m)$.
Step 2. For each triple of vertices $\{m_1, m_2, m_3\}$ of $G \setminus \{a_1, a_2, a_3, b_1, b_2, b_3\}$, if the three paths $P_1(m_1), P_2(m_2), P_3(m_3)$ exist and their vertices induce an even prism, then return this prism and stop.
If no 6-tuple yields an even prism, return the negative answer.

Complexity: $O(|V(G)|^9)$.

Proof of correctness. If the algorithm returns an even prism, then clearly G contains this prism. Conversely, suppose that G contains an even prism. Let K be a smallest even prism, and let $(a_1, a_2, a_3, b_1, b_2, b_3, m_1, m_2, m_3)$ be the frame of K. When the algorithm considers the 6-tuple $\{a_1, a_2, a_3, b_1, b_2, b_3\}$ and vertex m_1, it will find paths $R_1(m_1)$ and $S_1(m_1)$ since some paths in K do have the required properties. By two applications of Lemma 5.1, $P_1(m_1)$ is a chordless path from a_1 to b_1. A similar property holds for $P_2(m_2)$ and $P_3(m_3)$. By six applications of Lemma 5.1, these three paths do form an even prism of G. So the algorithm will detect this subgraph.

Complexity analysis. The number of 6-tuples $\{a_1, a_2, a_3, b_1, b_2, b_3\}$ to be tested is $O(|V(G)|^6)$. Given a 6-tuple, for each vertex m, finding the two paths $R_1(m), S_1(m)$, and testing whether their union $P_1(m)$ is a chordless path, takes time $O(|V(G)|^2)$. So Step 1 can be done in time $O(|V(G)|^3)$. Step 2 can be implemented as follows (as in [6]). Say that a pair of vertices $\{m_1, m_2\}$ is $(1, 2)$-*good* if the paths $P_1(m_1), P_2(m_2)$ are disjoint and have no edge between them except a_1b_1 and a_2b_2. For a given m_1, one can find all m_2 such that $\{m_1, m_2\}$ is a $(1, 2)$-good pair in time $O(|V(G)|^2)$. First mark as forbidden all the vertices that lie in or have a neighbor in $P_1(m_1)$. This takes time $O(|V(G)|^2)$. Then for every m_2, check whether $P_2(m_2)$ contains a forbidden vertex. For a given m_2, this take time $O(|V(G)|)$. Thus all $(1, 2)$-good pairs can be found in time $O(|V(G)|^3)$. Repeat this for $(1, 3)$-good pairs and $(2, 3)$-good pairs. Finally, for every triple $\{m_1, m_2, m_3\}$ check in constant time if the pairs $\{m_i, m_j\}$ are (i, j)-good for all $1 \le i < j \le 3$. This takes time $O(|V(G)|^3)$. Thus the overall complexity is time $O(|V(G)|^9)$.

6. Line-graphs of subdivisions of K_4.

The *line-graph* of a graph R is the graph whose vertices are the edges of R and where two vertices are adjacent if the corresponding edges of R have a common endvertex. *Subdividing* an edge xy in a graph means replacing it by a path of length at least two. A *subdivision* of a graph R is any graph obtained by repeatedly subdividing edges. Berge graphs that do not contain the line-graph of a bipartite subdivision of K_4 play an important role in the proof of the strong perfect graph theorem [7]. Thus recognizing them may be of

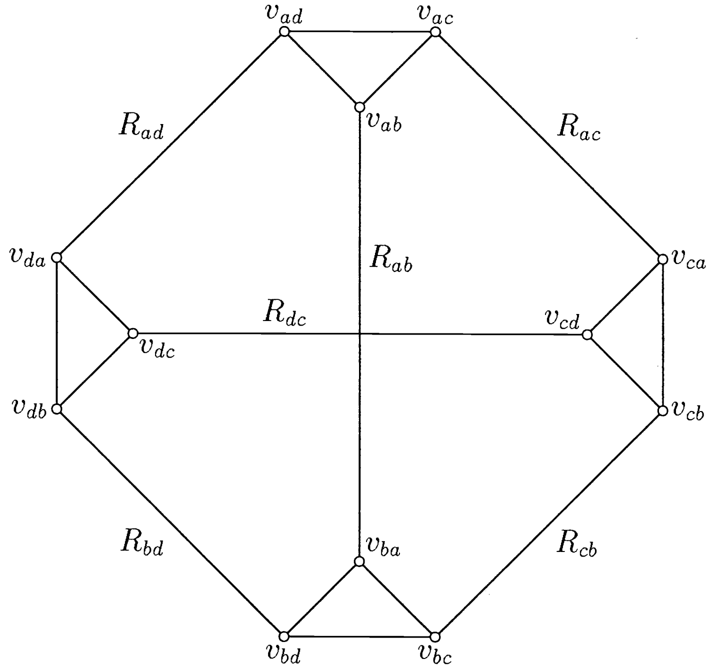


FIG. 2. *Line-graph of a subdivision of K_4.*

interest on its own. Moreover, solving this question is also useful for later use in the recognition of graphs in the class $\mathcal{A}'$ (see section 7). Again it turns out that deciding if a graph contains the line-graph of a subdivision of K_4 is NP-complete in general; see section 8.

We will first deal with subdivisions of K_4 that are not necessarily bipartite but are not too trivial in the following sense. Say that a subdivision of K_4 is *proper* if at least one edge of the K_4 is subdivided. It is easy to see that the line-graph of a subdivision of K_4 is proper if and only if it has a vertex that lies in only one triangle. If F is the line-graph of a proper subdivision R of K_4, let us denote by a, b, c, d the four vertices of K_4, i.e., the vertices of degree 3 in R. Then the three edges incident to each vertex $x \in \{a, b, c, d\}$ form a triangle in F, which will be labeled T_x and called a *basic* triangle of F. (F may have as many as two more, nonbasic, triangles.) In F there are six paths, each path being between vertices x, y of distinct basic triangles of F (and so this path can be labeled R_{xy} accordingly). Note that $R_{xy} = R_{yx}$, and the six distinct paths are vertex disjoint. Some of these paths may have length 0. In the basic triangle T_x, we denote by v_{xy} the vertex that is the end of the path R_{xy}. Thus F has paths R_{ab}, R_{ac}, R_{ad}, R_{bc}, R_{bd}, R_{cd}, and the vertices of the basic triangles of F are v_{ab}, v_{ac}, v_{ad}, v_{ba}, v_{bc}, v_{bd}, v_{ca}, v_{cb}, v_{cd}, v_{da}, v_{db} and v_{dc}. The graph F has no other edge than those in the four basic triangles and those in the six paths. See Figure 2.

For each of the six paths R_{xy} of F, we call m_{xy} one vertex that is roughly in the middle of R_{xy}, so that if α denotes the length of v_{xy}-R_{xy}-m_{xy} and β denotes the length of m_{xy}-R_{xy}-v_{yx}, then $\alpha - \beta \in \{-1, 0, 1\}$. Paths R_{xy} are called the *rungs* of F; vertices v_{xy} are called the *corners* of F; and the 18-tuple $(v_{ab}, v_{ac}, \ldots, v_{cd}, m_{ab}, \ldots, m_{cd})$ is called a *frame* of F.

Lemma 6.1. *Let G be a graph that contains no pyramid. Let F be an induced subgraph of G that is the line-graph of a proper subdivision of K_4 and F has smallest size with this property, and let $(v_{ab}, v_{ac}, \dots, v_{cd}, m_{ab}, \dots, m_{cd})$ be a frame of F. Let P be a path from v_{ab} to m_{ab} such that the interior vertices of P are not adjacent to any corner of F other than v_{ab} and P is a shortest path with these properties. Then $(V(F) \setminus V(R_{ab})) \cup V(P)$ induces the line-graph of a proper subdivision of K_4 of smallest size.*

Proof. Put $F' = F \setminus R_{ab}$. If v_{ab}, m_{ab} are equal or adjacent, then $P = v_{ab}$-R_{ab}-m_{ab} and the conclusion is immediate. So we may assume that v_{ab}, m_{ab} are distinct and not adjacent, which also implies $m_{ab} \neq v_{ba}$.

Claim 1. *If the interior vertices of P have no neighbor in F', then the lemma holds.*

Proof. Let u be the vertex of v_{ab}-P-m_{ab} that has neighbors in m_{ab}-R_{ab}-v_{ba} and is closest to v_{ab}. Let u' be the neighbor of u in m_{ab}-R_{ab}-v_{ba} closest to v_{ba}. Then v_{ab}-P-u-u'-R_{ab}-v_{ba} is a chordless path R, and $V(F') \cup V(R)$ induces the line-graph of a proper subdivision of K_4. So this subgraph has size at most the size of F, which is possible only if $u = m_{ab}$, and in this case $V(F') \cup V(R)$ induces the line-graph of a proper subdivision of K_4 of smallest size, so the lemma holds. □

Now we may assume that there exists a vertex $c_1 \in V(P)$ that has neighbors in F' and choose c_1 closest to v_{ab} along P. Also there exists a vertex $d_1 \in V(P)$ that has neighbors in F' and is chosen closest to m_{ab} along P. Let us show that this leads to a contradiction. See Figure 3.

Claim 2.

1. *The set $N(c_1) \cap V(F')$ consists of an edge of F'.*
2. *The set $N(d_1) \cap V(F')$ consists of an edge of F'.*

Proof. Call H the hole induced by $V(R_{ac}) \cup V(R_{bc}) \cup V(R_{bd}) \cup V(R_{ad})$.

First suppose that c_1 has no neighbor on H. Therefore, c_1 has neighbors in the interior of R_{cd}. Let c_2, c_3 be the neighbors of c_1, respectively, closest to v_{cd} and to v_{dc} along R_{cd}. If $c_2 = c_3$, the three paths c_2-c_1-P-v_{ab}, c_2-R_{cd}-v_{cd}-v_{ca}-R_{ca}-v_{ac}, c_2-R_{cd}-v_{dc}-v_{da}-R_{ad}-v_{ad} form a pyramid with triangle $\{v_{ab}, v_{ac}, v_{ad}\}$ and apex c_2, a contradiction. If c_2, c_3 are distinct and not adjacent, the three paths c_1-P-v_{ab}, c_1-c_2-R_{cd}-v_{cd}-v_{ca}-R_{ca}-v_{ac}, c_1-c_3-R_{cd}-v_{dc}-v_{da}-R_{ad}-v_{ad} form a pyramid with triangle $\{v_{ab}, v_{ac}, v_{ad}\}$, and apex c_1, a contradiction. If c_2, c_3 are adjacent, we have item 1 of the claim.

Now suppose that c_1 has neighbors on H. Define two chordless subpaths of H: $H_{ac} = H \setminus v_{ad}$ and $H_{ad} = H \setminus v_{ac}$. Let c_2 be the neighbor of c_1 on H_{ac} closest to v_{ac}, and let c_3 be the neighbor of c_1 on H_{ad} closest to v_{ad}. If $c_2 = c_3$, then $V(H) \cup V(c_1$-P-$v_{ab})$ induces a pyramid with triangle $\{v_{ab}, v_{ac}, v_{ad}\}$ and apex c_2, a contradiction. Therefore, $c_2 \neq c_3$. If c_2, c_3 are not adjacent, then the three paths c_1-P-v_{ab}, c_1-c_2-H_{ac}-v_{ac}, and c_1-c_3-H_{ad}-v_{ad} form a pyramid with triangle $\{v_{ab}, v_{ac}, v_{ad}\}$ and apex c_1, a contradiction. So c_2, c_3 are adjacent and are the only neighbors of c_1 on H. Up to symmetry, and by the definition of R, we may assume that c_2, c_3 are in the interior of R_{ac} or R_{bc}, because P was chosen so that no corner of F has a neighbor in it. If c_1 has no neighbor on R_{cd}, then conclusion 1 holds. Suppose that c_1 has a neighbor c_4 on R_{cd} and c_4 is closest to v_{dc}. Then the three paths c_1-P-v_{ab}, c_1-c_4-R_{cd}-v_{dc}-v_{da}-R_{da}-v_{ad}, c_1-c_2-H_{ac}-v_{ac} form a pyramid with triangle $\{v_{ab}, v_{ac}, v_{ad}\}$ and apex c_1, a contradiction. This completes the proof of item 1.

The proof of item 2 is similar, with the following adjustment: whenever path c_1-P-v_{ab} was used for item 1, we can use for item 2 a chordless path from d_1 to v_{ba} contained in d_1-P-m_{ab}-R_{ab}-v_{ba}. This completes the proof of the claim. □

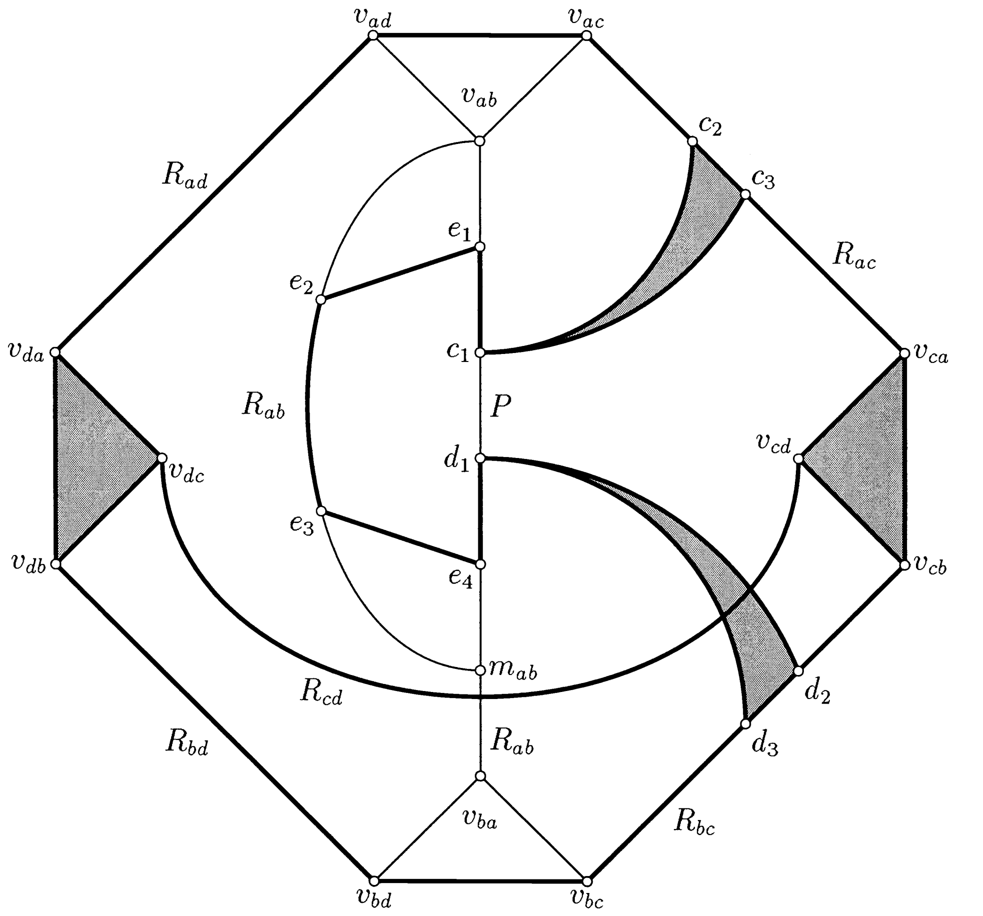


FIG. 3. *F and P for the proof of Lemma* 6.1.

CLAIM 3. *If J is the line-graph of a subdivision of K_4 with $V(J) \subseteq V(F') \cup V(P)$ and c_1 is a corner of J, then J is the line-graph of a proper subdivision of K_4.*

Proof. This claim follows immediately from the fact that c_1 belongs to exactly one triangle of J. □

In view of Claim 2, let c_2, c_3 be the two neighbors of c_1 in F' and d_2, d_3 be the two neighbors of d_1 in F', with $c_2c_3, d_2d_3 \in E(G)$.

CLAIM 4. *We may assume that c_2, c_3 lie in R_{ac} and d_2, d_3 in R_{cb} or R_{bd}.*

Proof. Recall from the definition of P that c_2, c_3, d_2, d_3 cannot be corners of F. If c_2c_3 is an edge of R_{cd}, then $V(v_{ab}\text{-}P\text{-}c_1) \cup V(H) \cup V(R_{cd})$ induces the line-graph of a subdivision of K_4, which is proper by Claim 3 and is strictly smaller than F, a contradiction. If c_2c_3 is an edge of R_{bc}, then $V(v_{ab}\text{-}P\text{-}c_1) \cup V(F')$ induces the line-graph of a subdivision of K_4, which is proper by Claim 3 and is strictly smaller than F, a contradiction. So c_2c_3 is an edge of R_{ac} or R_{ad}. Similarly we may assume that d_2d_3 is an edge of R_{bc} or R_{bd}. Then by symmetry the claim holds. □

We may assume that $v_{ac}, c_2, c_3, v_{ca}, d_2, d_3, v_{ad}$ appear in this order along H.

CLAIM 5. *Vertices c_1, d_1 are distinct and not adjacent.*

Proof. By Claims 2 and 4, we know that c_1, d_1 are distinct. If they are adjacent, the set $V(F') \cup \{c_1, d_1\}$ induces the line-graph of a subdivision of K_4, which is proper by Claim 3 and is strictly smaller than F, a contradiction. □

Let e_1 be the vertex of c_1-P-v_{ab} that has a neighbor e_2 in the interior of m_{ab}-R_{ab}-v_{ab} and is closest to c_1. Let e_4 be the vertex of d_1-P-m_{ab} that has a neighbor e_3 in the interior of m_{ab}-R_{ab}-v_{ab} and is closest to d_1. Given e_1, e_4, take e_2, e_3 as close to each other as possible along R_{ab}.

CLAIM 6. $e_1 \neq v_{ab}$.

Proof. Suppose $e_1 = v_{ab}$. Then the three paths v_{ab}-P-c_1, v_{ab}-v_{ac}-R_{ac}-c_2, v_{ab}-R_{ab}-e_3-e_4-P-d_1-d_2-H_{ac}-c_3 form a pyramid with triangle $\{c_1, c_2, c_3\}$ and apex v_{ab}, a contradiction. □

At this point we have obtained that c_1-P-e_1-e_2-R_{ab}-e_3-e_4-P-d_1 is a chordless path R whose interior vertices have no neighbor in F'. Moreover, the subgraph F_R induced by $V(F') \cup V(R)$ is the line graph of a subdivision of K_4, and it is proper by Claim 3.

CLAIM 7. $|V(F_R)| < |V(F)|$.

Proof. We need only show that the total length of the rungs of F_R is strictly smaller than the total length of the rungs of F. Let α be the length of v_{ab}-R_{ab}-m_{ab}, let β be the length of v_{ba}-R_{ab}-m_{ab}, and let δ be the number of those edges of F' that belong to the rungs of F.

The total length l of the rungs of F is equal to $\alpha + \beta + \delta = 2\alpha - \varepsilon + \delta$, with $\varepsilon = \alpha - \beta \in \{-1, 0, 1\}$ by the definition of m_{ab}.

The total length l_R of the rungs of F_R is at most $\delta + 2\alpha - 3$, and it is equal to this value only in the following case: $e_4 = m_{ab}$, there is only one vertex of R_{ab} between c_1 and d_1, $e_1 v_{ab} \in E(G)$, $e_2 v_{ab} \in E(G)$, and the paths P and v_{ab}-R_{ab}-m_{ab} have the same length. Indeed in this case the length of the rung of F_R whose ends are c_1, d_1 is equal to $2\alpha - 3$.

Thus in either case we have $l_R < l$ and the claim holds. □

Now the preceding claim leads to a contradiction, which proves the lemma. □

Lemma 6.1 is the basis of an algorithm for deciding if a graph contains a pyramid or the line-graph of a proper subdivision of K_4.

ALGORITHM 5. *Detection of the line-graph of a proper subdivision of K_4 in a graph that contains no pyramid.*

Input: *A graph G that contains no pyramid.*

Output: *An induced subgraph of G that is the line-graph of a proper subdivision of K_4, if G contains any; else the negative answer "G does not contain the line-graph of a proper subdivision of K_4."*

Method: *For every 12-tuple of vertices $T = (v_{ab}, v_{ac}, \ldots, v_{dc})$ such that each of $\{v_{ab}, v_{ac}, v_{ad}\}$, $\{v_{ba}, v_{bc}, v_{bd}\}$, $\{v_{ca}, v_{cb}, v_{cd}\}$, $\{v_{da}, v_{db}, v_{dc}\}$ induces a triangle, do:*

Step 1. For $i, j \in \{a, b, c, d\}$, $i < j$, compute the set F_{ij} of those vertices that are not adjacent to the vertices of T except possibly to v_{ij}, v_{ji}; for each $m \in V(G) \setminus T$, look for a shortest path $P_{ij}(m)$ from v_{ij} to m whose interior vertices are in F_{ij}, and look for a shortest path $Q_{ij}(m)$ from m to v_{ji} whose interior vertices are in F_i. If $P_{ij}(m)$ and $Q_{ij}(m)$ exist and their union is a chordless path from v_{ij} to v_{ji}, then call this path $R_{ij}(m)$.

Step 2. For each 6-tuple of vertices $\{m_{ab}, \ldots, m_{cd}\}$ of $G \setminus T$, if the six paths $P_{ab}(m_{ab}), \ldots, P_{cd}(m_{cd})$ exist and their vertices induce the line-graph of a proper subdivision of K_4, then return this subgraph and stop.

If no 12-tuple yields the line-graph of a proper subdivision of K_4, return the negative answer.

Complexity: $O(|V(G)|^{18})$.

Proof of correctness. If the algorithm returns the line-graph of a proper subdivision of K_4, then clearly G contains this subgraph. Conversely, suppose that G contains the line-graph of a proper subdivision of K_4. Let F be a smallest such subgraph, and let $(v_{ab}, \dots, v_{dc},\ m_{ab}, \dots, m_{cd})$ be a frame of F. When the algorithm considers the 12-tuple $(v_{ab}, \dots, v_{dc})$ and vertex m_{ab}, it will find paths $P_{ab}(m_{ab})$ and $Q_{ab}(m_{ab})$ since some paths in F do have the required properties. By two applications of Lemma 6.1, $R_{ab}(m_{ab})$ is a chordless path from v_{ab} to v_{ba}. A similar property holds for $R_{ac}(m_{ac})$, $\dots$, $R_{cd}(m_{cd})$. By 12 applications of Lemma 6.1, the vertices of these six paths do induce the line-graph of a proper subdivision of K_4. So the algorithm will detect this subgraph.

Complexity analysis. The number of 12-tuples to be tested is $O(|V(G)|^{12})$. Given a 12-tuple, for each pair $i, j \in \{a, b, c, d\}$ and each vertex m, finding the two paths $P_{ij}(m)$ and $Q_{ij}(m)$, and testing whether their union $R_{ij}(m)$ is a chordless path, takes time $O(|V(G)|^2)$. So Step 1 can be done in time $O(|V(G)|^3)$. Step 2 can be implemented as follows. Say that a pair of vertices $\{m_{ab}, m_{ac}\}$ is (ab, ac)-*good* if the paths $R_{ab}(m_{ab}), R_{ac}(m_{ac})$ are disjoint and have no edge between them except for $v_{ab}v_{ac}$; and say that a pair of vertices $\{m_{ab}, m_{cd}\}$ is (ab, cd)-*good* if the paths $R_{ab}(m_{ab}), R_{cd}(m_{cd})$ are disjoint and have no edge at all between them. For a given m_{ab}, one can find all m_{ac} such that $\{m_{ab}, m_{ac}\}$ is an (ab, ac)-good pair in time $O(|V(G)|^2)$. First mark as forbidden all the vertices that lie in or have a neighbor in $R_{ab}(m_{ab})$. This takes time $O(|V(G)|^2)$. Then for every m_{ac}, check whether $R_{ac}(m_{ac})$ contains no forbidden vertex. For a given m_{ac}, this takes time $O(|V(G)|)$. Thus all (ab, ac)-good pairs can be found in time $O(|V(G)|^3)$. Repeat this for (ab, ad)-good pairs, etc., and similarly for all (ab, cd)-good pairs, (ac, bd)-good pairs, and (ad, bc)-good pairs.

Finally, for every 6-tuple $\{m_{ab}, m_{ac}, m_{ad}, m_{bc}, m_{bd}, m_{cd}\}$ check in constant time if two vertices m_{ij}, m_{kl} form an (ij, kl)-good pair for all $1 \le i < j \le 3$, $1 \le k < l \le 3$. This takes time $O(|V(G)|^6)$. Thus the overall complexity is time $O(|V(G)|^{18})$.

Let us now focus on finding line-graphs of *bipartite* subdivisions of K_4.

LEMMA 6.2. *Let R be a subdivision of K_4 and F be the line-graph of R. Then either $R = K_4$, or F contains an odd hole, or R is a bipartite subdivision of K_4.*

Proof. Suppose $R \neq K_4$. Call a, b, c, d the four vertices of the K_4 of which R is a subdivision (i.e., the vertices of degree 3 in R), and for $i, j \in \{a, b, c, d\}$ with $i \neq j$, call C_{ij} the subdivision of edge ij. Suppose that F contains no odd hole and R is not bipartite. Then R contains an odd cycle Z. This cycle must be a triangle, for otherwise $L(R)$ contains an odd hole, a contradiction. So we may assume up to symmetry that a, b, c induce a triangle. Since $R \neq K_4$, we may assume that C_{ad} has length at least 2. But then one of $E(C_{ad}) \cup \{ac\} \cup E(C_{cd})$ or $E(C_{ad}) \cup \{ab\} \cup \{bc\} \cup C_{cd}$ is the edge set of an odd cycle of R, of length at least 5, so $L(R)$ contains an odd hole, a contradiction. □

Now we can devise an algorithm that decides if a graph with no odd hole contains the line-graph of a bipartite subdivision of K_4. This algorithm is simply Algorithm 5 applied to graphs that contain no odd hole, by the preceding lemma.

7. Recognition of graphs in class $\mathcal{A}'$.

To decide if a graph is in class $\mathcal{A}'$, it suffices to decide separately if it is Berge, if it has an antihole of length at least 5, and if it contains an odd prism. But again it turns out that this third question—deciding if a graph contains an odd prism—is NP-complete (see section 8). However, we can decide in polynomial time if a graph with no odd hole contains an odd prism. For this purpose the next lemmas will be useful.

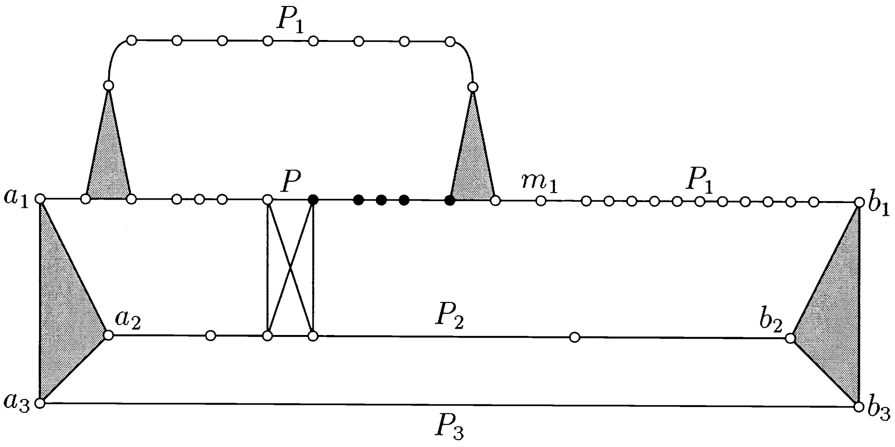


FIG. 4. *A graph with six odd prisms.*

LEMMA 7.1. *Let F be the line-graph of a bipartite subdivision of K_4. Then F contains an odd prism.*

Proof. Let R be a bipartite subdivision of K_4 such that F is the line-graph of R, and let a, b, c, d be the four vertices of degree 3 in R. We may suppose without loss of generality that a, b lie on the same side of the bipartition of R. Thus edge ab is subdivided to a path R_{ab} of even length, with the usual notation. Now it is easy to see that $F \setminus V(R_{cd})$ is an odd prism. □

Before we present an algorithm for recognizing graphs in class $\mathcal{A}'$, we can remark that the technique which worked well for detecting even prisms tends to fail for odd prisms. The graph featured in Figure 4 illustrates this problem. This graph G is the line-graph of a bipartite graph, so it is a Berge graph. For any two gray triangles, there exists one (and only one) odd prism that contain these two triangles. Moreover, the paths P_1, P_2, P_3 form an odd prism of G of minimal size. Yet, replacing P_1 (or the path a_1-P_1-m_1) by a shortest path with the same ends does not produce an odd prism. Thus an algorithm that would be similar to the even prism testing algorithm presented above may work incorrectly. We note, however, that in this example the graph G contains the line-graph of a proper subdivision of K_4 (the subgraph obtained by forgetting the black vertices). The next lemma shows that this remark holds in general.

LEMMA 7.2. *Let G be a graph that contains no odd hole and no line-graph of a proper subdivision of K_4. Let H be a prism in G, with triangles $\{a_1, a_2, a_3\}$ and $\{b_1, b_2, b_3\}$, formed by paths P_1, P_2, P_3, where for $i = 1, 2, 3$ path P_i is from a_i to b_i. Let P be any chordless path from a_1 to b_1 whose interior vertices are not adjacent to a_2, a_3, b_2, b_3. Then the three paths P, P_2, P_3 form a prism of G of the same parity as H.*

Proof. Note that we are not assuming that H is a smallest prism or that P is a shortest path. If the interior vertices of P have no neighbor on $P_2 \cup P_3$, then the lemma holds. (Note that P has the same parity as P_2 and P_3 since G contains no odd hole.) So suppose that some interior vertex c_1 of P has neighbors on $P_2 \cup P_3$, and choose c_1 closest to a_1 along P. Define paths $H_2 = P_2 + P_3 \setminus \{a_3\}$ and $H_3 = P_2 + P_3 \setminus \{a_2\}$. For $i = 2, 3$, let c_i be the neighbor of c_1 closest to a_i along H_i.

If $c_2 = c_3$, then the three paths c_2-c_1-P-a_1, c_2-H_2-a_2, c_2-H_3-a_3 form a pyramid with triangle $\{a_1, a_2, a_3\}$ and apex c_2, so G contains an odd hole, a contradiction. Thus $c_2 \neq c_3$. If c_2, c_3 are not adjacent, then the three paths c_1-P-a_1, c_1-c_2-H_2-a_2, c_1-c_3-H_3-a_3 form a pyramid with triangle $\{a_1, a_2, a_3\}$ and apex c_1, a contradiction. Thus c_2, c_3 are adjacent. Up to symmetry and since c_1 is not adjacent to b_2, b_3, we may assume that c_2c_3 is an edge of P_2. If c_1, b_1 are adjacent, then the three paths c_1-b_1, c_1-c_3-P_2-b_2, c_1-P-a_1-a_3-P_3-b_3 form a pyramid with triangle $\{b_1, b_2, b_3\}$ and apex c_1, a contradiction. Thus c_1, b_1 are not adjacent. Let a'_1 be the neighbor of a_1 in P_1. Let d_1 be the vertex of a'_1-P-c_1 that has neighbors in P_1 and is closest to c_1. Let d_2, d_3 be the neighbors of d_1 along P_1 that are closest to a_1 and b_1, respectively.

If $d_2 = d_3$, then either $d_1 \neq a'_1$ or $d_2 = a_1$, and in either case the three paths d_2-d_1-P-c_1, d_2-P_1-a_1-a_2-P_2-c_2, d_2-P_1-b_1-b_2-P_2-c_3 form a pyramid with triangle $\{c_1, c_2, c_3\}$ and apex d_2, a contradiction. Thus, $d_2 \neq d_3$. If d_2, d_3 are not adjacent, then the three paths d_1-d_2-P_1-a_1, d_1-d_3-P_1-b_1-b_3-P_3-a_3, d_1-P-c_1-c_2-P_2-a_2 form a pyramid with triangle $\{a_1, a_2, a_3\}$ and apex d_1, a contradiction. Thus, d_2, d_3 are adjacent. Then the four triangles $\{a_1, a_2, a_3\}$, $\{b_1, b_2, b_3\}$, $\{c_1, c_2, c_3\}$, $\{d_1, d_2, d_3\}$ and the six paths P_3, a_2-P_2-c_2, a_1-P_1-d_2, b_2-P_2-c_3, b_1-P_1-d_3, c_1-P-d_1 form the line-graph of a subdivision of K_4, and it is not the line-graph of K_4 since $a_3 \neq b_3$; so G contains the line-graph of a proper subdivision of K_4, a contradiction. ∎

Now we can present an algorithm that decides if a graph with no odd hole contains an odd prism.

Algorithm 6. *Detection of an odd prism in a graph that contains no odd hole.*

Input: *A graph G that contains no odd hole.*

Output: *An odd prism induced in G, if G contains any, else the negative answer "G contains no odd prism."*

Method: *Using Algorithm 5, test whether G contains the line-graph of a proper subdivision of K_4. If G contains such a subgraph F, for each of the six rungs R of F, test if $F \setminus V(R)$ is an odd prism, and if it is, return this odd prism. If Algorithm 5 answers that G does not contain the line-graph of a proper subdivision of K_4, then for every 6-tuple $(a_1, a_2, a_3, b_1, b_2, b_3)$ do:*

For $i = 1, 2, 3$ compute a shortest path P_i from a_i to b_i whose interior vertices are not adjacent to a_{i+1}, a_{i+2}, b_{i+1}, and b_{i+2} (subscripts are understood modulo 3). If paths P_1, P_2, P_3 exist and form an odd prism, return this prism and stop.

If no 6-tuple has produced an odd prism, return the answer no.

Complexity: $O(|V(G)|^{18})$.

Proof of correctness. If G contains the line-graph of proper subdivision of K_4, this will be detected by Algorithm 5. If G contains no odd hole and no odd prism, then Lemma 7.1 ensures that G cannot contain the line-graph of a proper subdivision of K_4. So the algorithm will return the correct answer.

Now suppose that G does not contain the line graph of a proper subdivision of K_4 and G contains an odd prism, with triangles $\{a_1, a_2, a_3\}$ and $\{b_1, b_2, b_3\}$. Then in some step the algorithm will consider these six vertices, and it will find paths P_i since the corresponding paths of the prism have the required properties. By three applications of Lemma 7.2, we obtain that P_1, P_2, P_3 form an odd prism, and so the algorithm will detect it.

Complexity analysis. The complexity is clearly determined by its costliest step, which is Algorithm 5.

Now deciding if a graph is in class $\mathcal{A}'$ can be done as follows. Test if G contains an antihole of length at least 5 as explained earlier; test if G is Berge using the algorithm from section 2; then use Algorithm 6 to test if G contains no odd prism.

The complexity is the same as that of Algorithm 6.

We note that a conjecture stronger than Conjecture 1 was proposed in [8]: *If G is a graph in $\mathcal{A}'$ and G is not a clique, then G admits an even pair whose contraction yields a graph in $\mathcal{A}'$.* This stronger conjecture could actually be false even if Conjecture 1 is true. We would like to remark that if the stronger conjecture is true, then the algorithm for recognizing graphs in class $\mathcal{A}'$ can be used to color optimally the vertices of any graph $G \in \mathcal{A}'$ (even if a proof of the stronger conjecture is not algorithmic); this can be done similarly to the remark made at the end of section 4, as follows. Enumerate all pairs of nonadjacent vertices of G and test whether their contraction produces a graph in class $\mathcal{A}'$; the assumed validity of the stronger conjecture ensures that at least one pair will work. Then iterate this procedure until the contractions turn the graph into a clique. In terms of complexity, since we may need to check $O(|V(G)|^2)$ pairs at each contraction step, and there may be $O(|V(G)|)$ steps, we end up with total complexity $O(|V(G)|^{21})$. Thus it is desirable to find a proof of Conjecture 1 or of the stronger conjecture that produces an algorithm with lower complexity.

8. NP-complete problems. In this section we show that the following five problems are NP-complete:

1. Decide if a graph contains a prism.
2. Decide if a graph contains an even prism.
3. Decide if a graph contains an odd prism.
4. Decide if a graph contains the line-graph of a proper subdivision of K_4.
5. Decide if a graph contains the line-graph of a bipartite subdivision of K_4.

We have seen in the preceding sections that all these problems are polynomial when the input is restricted to the class of graphs that contain no odd hole.

The above NP-completeness results can all be derived from the following theorem. Let us call problem Π the decision problem whose input is a triangle-free graph G and two nonadjacent vertices a, b of G of degree 2 and whose question is, "Does G have a hole that contains both a, b?" Bienstock [5] mentions that this problem is NP-complete in general (i.e., not restricted to triangle-free graphs). We adapt his proof here for triangle-free graphs.

THEOREM 8.1. *Problem Π is NP-complete.*

Proof. Let us give a polynomial reduction from the problem 3-SATISFIABILITY of Boolean functions to problem Π. Recall that a Boolean function with n variables is a mapping f from $\{0,1\}^n$ to $\{0,1\}$. A Boolean vector $\xi \in \{0,1\}^n$ is a *truth assignment* for f if $f(\xi) = 1$. For any Boolean variable x in $\{0,1\}$, we write $\overline{x} := 1 - x$, and each of $x, \overline{x}$ is called a *literal*. An instance of 3-SATISFIABILITY is a Boolean function f given as a product of clauses, each clause being the Boolean sum $\vee$ of three literals; the question is whether f admits a truth assignment. The NP-completeness of 3-SATISFIABILITY is a fundamental result in complexity theory; see [10].

Let f be an instance of 3-SATISFIABILITY, consisting of m clauses $C_1, \dots, C_m$ on n variables $x_1, \dots, x_n$. Let us build a graph G_f with two specialized vertices a, b, such that there will be a hole containing both a, b in G_f if and only if there exists a truth assignment for f.

For each variable x_i ($i = 1, \dots, n$), make a graph $G(x_i)$ with eight vertices $a_i, b_i, t_i, f_i, a'_i, b'_i, t'_i, f'_i$, and 10 edges $a_it_i, a_if_i, b_it_i, b_if_i$ (so that $\{a_i, b_i, t_i, f_i\}$ induces a hole), $a'_it'_i, a'_if'_i, b'_it'_i, b'_if'_i$ (so that $\{a'_i, b'_i, t'_i, f'_i\}$ induces a hole), and $t_if'_i, t'_if_i$. See Figure 5.

For each clause C_j ($j = 1, \dots, m$), with $C_j = u_j^1 \vee u_j^2 \vee u_j^3$, where each u_j^p ($p = 1, 2, 3$) is a literal from $\{x_1, \dots, x_n, \overline{x}_1, \dots, \overline{x}_n\}$, make a graph $G(C_j)$ with five vertices

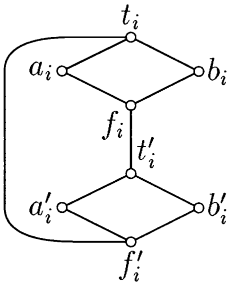


FIG. 5. *Graph $G(x_i)$.*

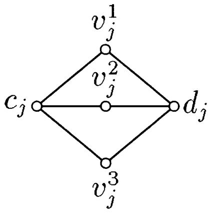


FIG. 6. *Graph $G(C_j)$.*

$c_j, d_j, v_j^1, v_j^2, v_j^3$ and six edges so that each of c_j, d_j is adjacent to each of v_j^1, v_j^2, v_j^3. See Figure 6. For $p = 1, 2, 3$, if $u_j^p = x_i$ then add two edges $v_j^p f_i, v_j^p f'_i$, while if $u_j^p = \overline{x}_i$ then add two edges $v_j^p t_i, v_j^p t'_i$. See Figure 7.

The graph G_f is obtained from the disjoint union of the $G(x_i)$'s and the $G(C_j)$'s as follows. For $i = 1, \ldots, n-1$, add edges $b_i a_{i+1}$ and $b'_i a'_{i+1}$. Add an edge $b'_n c_1$. For $j = 1, \ldots, m-1$, add an edge $d_j c_{j+1}$. Introduce the two specialized vertices a, b and add edges aa_1, aa'_1 and bd_m, bb_n. See Figure 8. Clearly the size of G_f is polynomial (actually linear) in the size $n + m$ of f. Moreover, it is easy to see that G_f contains no triangle and that a, b are nonadjacent and both have degree 2.

Suppose that f admits a truth assignment $\xi \in \{0, 1\}^n$. We build a hole in G by selecting vertices as follows. Select a, b. For $i = 1, \ldots, n$, select a_i, b_i, a'_i, b'_i; moreover, if $\xi_i = 1$ select t_i, t'_i, while if $\xi_i = 0$ select f_i, f'_i. For $j = 1, \ldots, m$, since ξ is a truth assignment for f, at least one of the three literals of C_j is equal to 1, say, $u_j^p = 1$ for some $p \in \{1, 2, 3\}$. Then select c_j, d_j, and v_j^p. Now it is a routine matter to check that the selected vertices induce a cycle Z that contains a, b and that Z is chordless, so it is a hole. The main point is that there is no chord in Z between some subgraph $G(C_j)$ and some subgraph $G(x_i)$, for that would be either an edge $t_i v_j^p$ (or $t'_i v_j^p$) with $u_j^p = x_i$ and $\xi_i = 1$ or, symmetrically, an edge $f_i v_j^p$ (or $f'_i v_j^p$) with $u_j^p = \overline{x}_i$ and $\xi_i = 0$, in either case a contradiction to the way the vertices of Z were selected.

Conversely, suppose that G_f admits a hole Z that contains a, b. Clearly Z contains a_1, a'_1 since these are the only neighbors of a in G_f.

CLAIM 8. *For $i = 1, \ldots, n$, Z contains exactly six vertices of $G(x_i)$: four are a_i, a'_i, b_i, b'_i and the other two are either t_i, t'_i or f_i, f'_i.*

Proof. First we prove the claim for $i = 1$. Since a, a_1 are in Z and a_1 has only three neighbors a, t_1, f_1, exactly one of t_1, f_1 is in Z. Likewise, exactly one of t'_1, f'_1 is in Z. If t_1, f'_1 are in Z, then the vertices a, a_1, a'_1, t_1, f'_1 are all in Z and they induce a hole that does not contain b, a contradiction. Likewise, we do not have both t'_1, f_1 in Z. Therefore, up to symmetry we may assume that t_1, t'_1 are in Z and f_1, f'_1 are not.

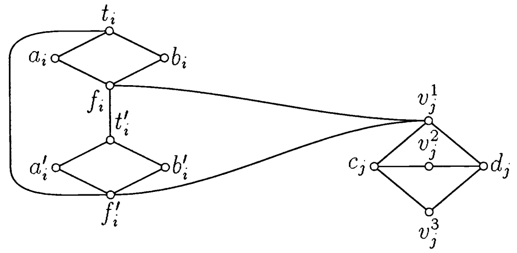


FIG. 7. *The two edges added to G_f in the case $u_j^1 = x_i$.*

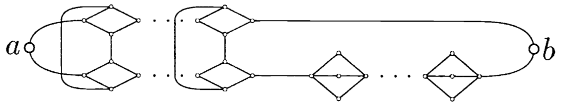


FIG. 8. *Graph G_f.*

If a vertex v_j^p of some $G(C_j)$ $(1 \leq j \leq m,\ 1 \leq p \leq 3)$ is in Z and is adjacent to t_1, then, since this v_j^p is also adjacent to t'_1, we see that the vertices $a, a_1, a'_1, t_1, t'_1, v_j^p$ are all in Z and induce a hole that does not contain b, a contradiction. Thus the neighbor of t_1 in $Z \setminus a_1$ is not in any $G(C_j)$ $(1 \leq j \leq m)$, so that neighbor is b_1. Likewise b'_1 is in Z. So the claim holds for $i = 1$. Since b_1 is in Z and exactly one of t_1, f_1 is in Z, and b_1 has degree 3 in G_f, we obtain that a_2 is in Z, and similarly a'_2 is in Z. Now the proof of the claim for $i = 2$ is essentially the same as for $i = 1$, and by induction the claim holds up to $i = n$. □

CLAIM 9. *For $j = 1, \ldots, m$, Z contains c_j, d_j, and exactly one of v_j^1, v_j^2, v_j^3.*

Proof. First we prove this claim for $j = 1$. By Claim 8, b'_n is in Z and exactly one of t'_n, f'_n is in Z, so (since b'_n has degree 3 in G_f) c_1 is in Z. Consequently, exactly one of v_1^1, v_1^2, v_1^3 is in Z, say, v_1^1. The neighbor of v_1^1 in $Z \setminus c_1$ cannot be a vertex of some $G(x_i)$ $(1 \leq i \leq n)$, for that would be either t_i (or f_i) and thus, by Claim 8, t'_i (or f'_i) would be a third neighbor of v_1^1 in Z, a contradiction. Thus the other neighbor of v_1^1 in Z is d_1, and the claim holds for $j = 1$. Since d_1 has degree 4 in G_f and exactly one of v_1^1, v_1^2, v_1^3 is in Z, it follows that its fourth neighbor c_2 is in Z. Now the proof of the claim for $j = 2$ is the same as for $j = 1$, and by induction the claim holds up to $j = m$. □

We can now make a Boolean vector ξ as follows. For $i = 1, \ldots, n$, if Z contains t_i, t'_i set $\xi_i = 1$; if Z contains f_i, f'_i set $\xi_i = 0$. By Claim 8 this is consistent. Consider any clause C_j $(1 \leq j \leq m)$. By Claim 9 and up to symmetry we may assume that v_j^1 is in Z. If $u_j^1 = x_i$ for some $i \in \{1, .., n\}$, then the construction of G_f implies that f_i, f'_i are not in Z, so t_i, t'_i are in Z, so $\xi_i = 1$, so clause C_j is satisfied by x_i. If $u_j^1 = \overline{x}_i$ for some $i \in \{1, .., n\}$, then the construction of G_f implies that t_i, t'_i are not in Z, so f_i, f'_i are in Z, so $\xi_i = 0$, so clause C_j is satisfied by $\overline{x}_i$. Thus ξ is a truth assignment for f. This completes the proof of the theorem. □

Now we can prove the main result of this section.

THEOREM 8.2. *The following problems are NP-complete:*

1. *Decide if a graph contains a prism.*
2. *Decide if a graph contains an odd prism.*
3. *Decide if a graph contains an even prism.*
4. *Decide if a graph contains the line-graph of a proper subdivision of K_4.*
5. *Decide if a graph contains the line-graph of a bipartite subdivision of K_4.*

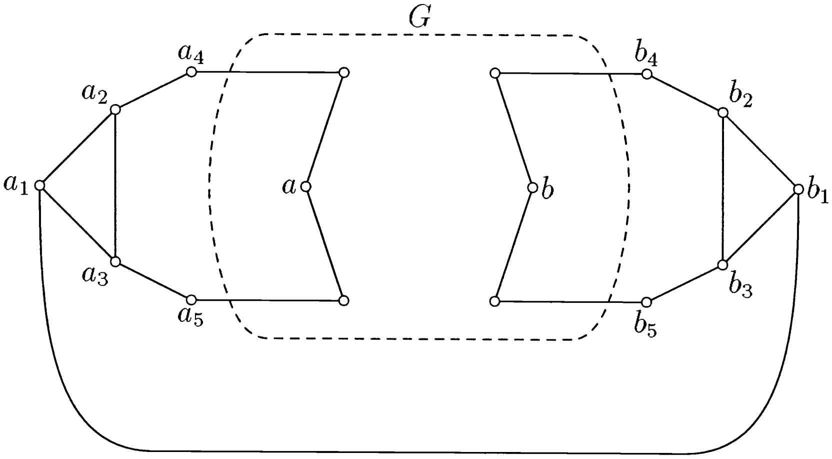


Fig. 9. *Problem* 1: *G and G'.*

Proof. For each of these five problems we show a reduction from problem Π to this problem. So let (G, a, b) be any instance of problem Π, where G is a triangle-free graph and a, b are nonadjacent vertices of G of degree 2. Let us call a', a'' the two neighbors of a and b', b'' the two neighbors of b in G.

Reduction to problem 1. Starting from G, build a graph G' as follows (see Figure 9). Replace vertex a by five vertices a_1, a_2, a_3, a_4, a_5 with five edges a_1a_2, a_1a_3, a_2a_3, a_2a_4, a_3a_5, and put edges a_4a' and a_5a''. Do the same with b, with five vertices named $b_1, \ldots, b_5$ instead of $a_1, \ldots, a_5$ and with the analogous edges. Add an edge a_1b_1. Since G has no triangle, G' has exactly two triangles $\{a_1, a_2, a_3\}$ and $\{b_1, b_2, b_3\}$. Moreover we see that G' contains a prism if and only if G contains a hole that contains a and b. Thus every instance of Π can be reduced polynomially to an instance of problem 1, which proves that problem 1 is NP-complete.

Reduction to problem 2. Starting from G, build the same graph G' as above. Then build four graphs $G_{i,j}$ $(i, j \in \{0, 1\})$ as follows. If $i = 1$, subdivide the edge a_2a_4 into a path of length 2; else do not subdivide it. Likewise, subdivide the edge a_3a_5 if and only if $j = 1$. Now G contains a hole that contains a and b if and only if at least one of the four graphs $G_{i,j}$ contains an odd prism. Thus every instance of Π can be reduced polynomially to four instances of problem 2.

Reduction to problem 3. Starting from G, build the four graphs $G_{i,j}$ as above and in each of them subdivide the edge a_1b_1. Then G contains a hole that contains a and b if and only if at least one of these four new graphs contains an even prism. Thus every instance of Π can be reduced polynomially to four instances of problem 3.

Reduction to problem 4. Starting from G, build a graph G'' as follows (see Figure 10). Remove vertices a and b and add 12 vertices v_{ab}, v_{ac}, v_{ad}, v_{ba}, v_{bc}, v_{bd}, v_{ca}, v_{cb}, v_{cd}, v_{da}, v_{db}, v_{dc}. Add edges such that each of $\{v_{ab}, v_{ac}, v_{ad}\}$, $\{v_{ba}, v_{bc}, v_{bd}\}$, $\{v_{ca}, v_{cb}, v_{cd}\}$, and $\{v_{da}, v_{db}, v_{dc}\}$ is a triangle. Add edges $v_{ab}v_{ba}$, $v_{dc}v_{cd}$, $v_{bd}v_{db}$, $v_{bc}v_{cb}$, $v_{ad}a'$, $v_{ac}a''$, $v_{da}b'$, $v_{ca}b''$. The graph G'' contains exactly four triangles, and G contains a hole through a and b if and only if G'' contains the line-graph of a proper subdivision of K_4. Thus every instance of Π can be reduced polynomially to an instance of problem 4.

Reduction to problem 5. Starting from G'', make four graphs $G''_{i,j}$ $(i, j \in \{0, 1\})$ as follows. If $i = 1$ subdivide the edge $v_{ad}a'$ into a path of length 2, else do not

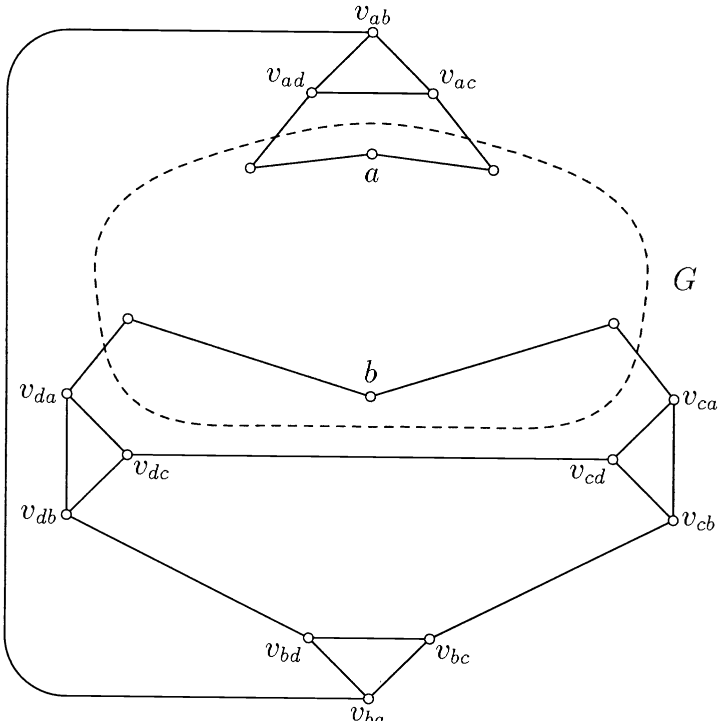


FIG. 10. *Problem* 4: *G and* G''.

subdivide it. Subdivide likewise the edge $v_{ac}a''$ if and only if $j = 1$. Now G contains a hole through a and b if and only if one of the four graphs $G''_{i,j}$ contains the line-graph of a bipartite subdivision of K_4. So every instance of Π can be reduced polynomially to four instances of problem 5. This completes the proof of the theorem. □

9. Conclusion.

We summarize the complexity results mentioned in this paper in the following table, whose columns correspond to the class of graphs taken as instances and whose lines correspond to the subgraph that we look for. The symbol n refers to the number of vertices of the input graph; 1 means trivial, NPC means NP-complete, and a question mark means unsolved.

	General graphs	Graphs with no pyramid	Graphs with no odd hole
Pyramid or prism	n^5	n^5	n^5
Pyramid	n^9 [6]	1	1
Prism	NPC	n^5	n^5
LGPSK_4	NPC	n^{18}	n^{18}
LGBSK_4	NPC	?	n^{18}
Odd prism	NPC	?	n^{18}
Even prism	NPC	?	n^9

Acknowledgements. We thank Bruce Reed for many interesting remarks which led to shortening and simplifying some of our proofs. We are grateful to the referees for their careful reading and to one referee who suggested an improvement in the complexity of Algorithms 4 and 5.

REFERENCES

[1] C. Berge, *Les problèmes de coloration en théorie des graphes*, Publ. Inst. Stat. Univ. Paris, 9 (1960), pp. 23–160.

[2] C. Berge, *Färbung von Graphen, deren sämtliche bzw. deren ungerade Kreise starr sind (Zusammenfassung)*, Wiss. Z. Martin Luther Univ. Math.-Natur. Reihe (Halle-Wittenberg), 10 (1961), pp. 114–115.

[3] C. Berge, *Graphs*, North-Holland, Amsterdam, New York, 1985.

[4] M. E. Bertschi, *Perfectly contractile graphs*, J. Combin. Theory Ser. B, 50 (1990), pp. 222–230.

[5] D. Bienstock, *On the complexity of testing for even holes and induced odd paths*, Discrete Math., 90 (1991), pp. 85–92; corrigendum in Discete Math., 102 (1992), p. 109.

[6] M. Chudnovsky, G. Cornuéjols, X. Liu, P. Seymour, and K. Vušković, *Recognizing Berge graphs*, Combinatorica, 25 (2005), pp. 143–186.

[7] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas, *The strong perfect graph theorem*, Ann. Math., to appear.

[8] H. Everett, C. M. H. de Figueiredo, C. Linhares Sales, F. Maffray, O. Porto, and B. A. Reed, *Even pairs*, in Perfect Graphs, J. L. Ramírez-Alfonsín and B. A. Reed, eds., Wiley-Interscience, New York, 2001, pp. 67–92.

[9] J. Fonlupt and J. P. Uhry, *Transformations which preserve perfectness and h-perfectness of graphs*, Ann. Discete Math., 16 (1982), pp. 83–85.

[10] M. R. Garey and D. S. Johnson, *Computer and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman, San Fransisco, 1979.

[11] C. Linhares Sales, F. Maffray, and B. A. Reed, *On planar perfectly contractile graphs*, Graphs Combin., 13 (1997), pp. 167–187.

[12] F. Maffray and N. Trotignon, *A class of perfectly contractile graphs*, J. Combin. Theory Ser. B, to appear.

[13] J. L. Ramírez-Alfonsín and B. A. Reed, *Perfect Graphs*, Wiley-Interscience, New York, 2001.

[14] B. A. Reed, *Problem session on parity problems*, in DIMACS Workshop on Perfect Graphs, Princeton University, Princeton, NJ, 1993.

[15] N. Trotignon, *Graphes Parfaits: Structure et Algorithmes*, Ph.D. thesis, University of Grenoble, Grenoble, France, 2004.

SIAM J. Discrete Math.
Vol. 19, No. 3, pp. 575–587

ON THE 3-TERMINAL CUT POLYHEDRON*

MOHAMED DIDI BIHA†

Abstract. Given $G = (V, E)$ an undirected graph and $A = \{n_1, n_2, n_3\} \subseteq V$ a specified set of terminal nodes, a 3-terminal cut is a subset of edges whose removal disconnects each terminal from the rest. Given a nonnegative cost vector $c \in \mathbb{R}_+^{|E|}$, the optimal 3-terminal cut problem is to find a 3-terminal cut of minimum cost. In this paper we consider the polyhedron $\mathrm{LP}(G, A)$, the linear relaxation of the 3-terminal cuts polyhedron $\mathrm{P}(G, A)$. We give a characterization of the pairs (G, A) for which $\mathrm{LP}(G, A)$ is integer. This result was conjectured by Cunningham in [*Reliability of Compter and Communications Networks*, AMS, Providence, RI, 1991, pp. 105–120].

Key words. A-cut, polyhedron

AMS subject classifications. 90C27, 90C57

DOI. 10.1137/S0895480104445149

1. Introduction.

Let $G = (V, E)$ be an undirected graph and $A = \{n_1, \ldots, n_k\}$ be a set of k specified nodes or *terminals*. A *k-terminal cut*, or an *A-cut*, is a set of edges $F \subseteq E$ such that the removal of F of E disconnects each terminal from all the others. The *optimal A-cut problem* is, given G, A, and a cost vector $c \in \mathbb{R}_+^{|E|}$, to find an A-cut of minimum cost.

The optimal A-cut problem arises in the minimization of communication costs in parallel computing systems [7]. Other applications involve partitioning files among the nodes of the network, assigning users to base computers in a multicomputer environment, and partitioning the elements of a circuit into the subcircuits that go in different chips [5].

For any $k \geq 3$, the optimal k-terminal cut is NP-hard even if $c(e) = 1$ for all $e \in E$ [5]. For any fixed k the optimal k-terminal cut problem can be solved in polynomial time on planar graphs [5]. When $k = 2$ the problem is reduced to the maximum flow problem and thus can be solved in polynomial time.

In [1], Călinescu, Karloff, and Rabani gave a new linear programming relaxation for the optimal A-cut problem and an approximation algorithm having performance guarantee $1.5 - \frac{1}{k}$. For $|A| = 3$, Cunningham and Tang [4] gave an approximation algorithm for the optimal A-cut problem having performance guarantee $\frac{12}{11}$.

The present paper concerns only the optimal A-cut problem when $|A| = 3$.

If $G = (V, E)$ is a graph and $F \subseteq E$, the $0-1$ vector $x^F \in \mathbb{R}^{|E|}$ with $x^F(e) = 1$ if $e \in E$ and $x^F(e) = 0$ if not is called the *incidence vector* of F. Define the polyhedron

$$\mathrm{P}(G, A) = \mathrm{conv}\{x^F \mid F \text{ is an } A\text{−cut}\} + \mathbb{R}_+^{|E|}.$$

$\mathrm{P}(G, A)$ is called the *A-cut polyhedron*. Notice that it is the dominant of the convex hull of A-cut vectors. If $c \geq 0$ then the A-cut problem is equivalent to solving the

*Received by the editors July 13, 2004; accepted for publication (in revised form) February 6, 2005; published electronically November 15, 2005.
http://www.siam.org/journals/sidma/19-3/44514.html

†Laboratoire d'Analyse non Linéaire et Géométrie, Université d'Avignon, 339 chemin des Meinajaries, 84911 Avigon, France (mohamed.didi-biha@univ-avignon.fr).

linear program

$$\min \left\{ \sum_{e \in E} c(e)x(e),\ x \in \mathrm{P}(G, A) \right\}.$$

Chopra and Rao [2] and Cunningham [3] have studied the polyhedron $\mathrm{P}(G, A)$ and given various families of facet defining inequalities.

Let $G = (V, E)$ be a graph and a set $A = \{n_1, n_2, n_3\}$ of terminals. An *A-path* is the edge set of a simple path in G from one terminal to another. An *A-tree* is the edge set of a tree such that the set of degree-one nodes is a subset of A. Notice that an A-path is also an A-tree. For the rest, when we consider an A-tree $\mathcal{T}$ we suppose that the set of degree-one nodes is A (i.e., $\mathcal{T}$ is not an A-path). Given $w \in \mathbb{R}^{|E|}$ and $F \subseteq E$, $w(F)$ will denote $\sum_{e \in F} w(e)$. If F is an A-cut, then x^F must satisfy the following inequalities:

$$x(\mathcal{P}) \geq 1 \qquad \text{for each } A\text{-path } \mathcal{P}, \tag{1.1}$$

$$x(\mathcal{T}) \geq 2 \qquad \text{for each } A\text{-tree } \mathcal{T}. \tag{1.2}$$

The inequalities (3.1) are called *A-path inequalities* and the inequalities (3.2) are called *A-tree inequalities*. Define the polyhedron

$$\mathrm{LP}(G, A) = \{x \geq 0; x \text{ satisfies (3.1) and (3.2)}\}.$$

Chopra and Rao [2] and Cunningham [3] have shown that the A-cut problem can be formulated as

$$\begin{aligned} \min \quad & \sum_{e \in E} c(e)x(e) \\ \text{s.t.} \quad & x \in \mathrm{LP}(G, A), \\ & x \text{ integer}. \end{aligned}$$

The separation problem for inequalities (3.1) and (3.2) (i.e., the problem that consists of finding whether a given vector $x \in \mathbb{R}^{|E|}$ satisfies inequalities (3.1) and (3.2), and if not to find an inequality which is violated by x) can be solved in polynomial time [3]. This implies by the ellipsoid method [6] that the optimal A-cut problem can be solved in polynomial time on pairs (G, A) for which $\mathrm{LP}(G, A)$ is integer. In [3], Cunningham called these pairs *nice pairs*. In this paper we give a characterization of these pairs.

The paper is organized as follows. In the next section we give more notation and definitions, and we give some structural properties of the extreme points of $\mathrm{LP}(G, A)$. In section 3 we give a characterization of the nice pairs (G, A). Then in the fourth section we shall prove the main theorem of this paper.

The remainder of this section is devoted to more definitions and notation.

The graphs we consider are finite and undirected. We denote a graph by $G = (V, E)$, where V is the node set and E is the edge set. If e is an edge with end-nodes u and v, then we write $e = (uv)$. If $W \subseteq V$, the set of edges having one end-node in W and the other one in $\overline{W} = V \setminus W$ is called a *cut* and is denoted by $\delta(W)$. If $W = \{v\}$ for some $v \in V$, then we write $\delta(v)$ for $\delta(W)$. The set of edges having both end-nodes in W will be denoted $E(W)$. If W_1, W_2 are disjoint subsets of V, then $[W_1, W_2]$ denotes the set of edges of G which have one node in W_1 and the other one in W_2. For $U \subseteq V$ we denote by $G(U)$ the induced subgraph on U (i.e., $G(U) = (U, E(U))$. If $F \subseteq E$, then $V(F)$ denotes the set of nodes of F and $G(F)$ the subgraph of G induced by F.

2. Preliminaries. Let $G=(V,E)$ be a graph and $A=\{n_1,n_2,n_3\}$ be a set of terminal nodes. Let X be an extreme point of $\mathrm{LP}(G,A)$ and $E_0(X)$ be the set of edges e such that $X(e)=0$. Define

$$\mathcal{P}(X)=\{\mathcal{P} \text{ an } A\text{−path} \mid X(\mathcal{P})=1\} \text{ and } \mathcal{T}(X)=\{\mathcal{T} \text{ an } A\text{−tree} \mid X(\mathcal{T})=2\}.$$

Since X is an extreme point of $\mathrm{LP}(G,A)$, there must exist $\mathcal{P}^*(X)\subseteq\mathcal{P}(X)$ and $\mathcal{T}^*(X)\subseteq\mathcal{T}(X)$ such that X is the unique solution of the system $S(X)$ defined by

$$S(X)\begin{cases} x(e)=0 & \forall\, e\in E_0(X),\\ x(\mathcal{P})=1 & \forall\, \mathcal{P}\in\mathcal{P}^*(X),\\ x(\mathcal{T})=2 & \forall\, \mathcal{T}\in\mathcal{T}^*(X),\end{cases}$$

where $|E_0(X)|+|\mathcal{P}^*(X)|+|\mathcal{T}^*(X)|=|E|$.

We have the following lemmas.

Lemma 2.1. *$X(e)\le 1$ for all $e\in E$.*

Proof. Suppose that there exists $e_0\in E$ such that $X(e_0)>1$. Since no path $\mathcal{P}\in\mathcal{P}(X)$ contains e_0, there must exist $\mathcal{T}\in\mathcal{T}^*(X)$ such that $e_0\in\mathcal{T}$. Let $\mathcal{P}$ be the only A-path in $G(\mathcal{T})$ from n_1 to n_2. Without loss of generality (w.l.o.g.), we can suppose that $e_0\notin\mathcal{P}$. As $X(\mathcal{P})+X(e_0)\le X(\mathcal{T})=2$ and $X(e_0)>1$, we have $X(\mathcal{P})<1$. But this contradicts the fact that $X\in\mathrm{LP}(G,A)$. □

Lemma 2.2. *Let $G=(V,E)$ be a graph, $A=\{n_1,n_2,n_3\}$ be a terminal set, and X be an extreme point of LP(G,A). Suppose that there exists $f\in E$ such that $X(f)=1$. Then X^*, the restriction of X to the graph $G^*=(V,E\setminus\{f\})$, is an extreme point of LP(G^*,A).*

Proof. Assume that X^* is not an extreme point of $\mathrm{LP}(G^*,A)$. Let $S^*(X)$ be the system obtained from $S(X)$ by deleting all the equalities containing $x(f)$ with a nonzero coefficient. There must exist an extreme point Y of $\mathrm{LP}(G^*,A)$ such that Y is a solution of $S^*(X)$. Let $\bar{X}\in\mathbb{R}^{|E|}$ be the following solution:

$$\bar{X}(e)=\begin{cases} Y(e) & \text{if } e\in E\setminus\{f\},\\ 1 & \text{if } e=f.\end{cases}$$

Let $\mathcal{P}\in\mathcal{P}^*(X)$ such that $f\notin\mathcal{P}$. Since the system $S^*(X)$ contains the equation $x(\mathcal{P})=1$, we have $\bar{X}(\mathcal{P})=Y(\mathcal{P})=1$. Similarly, we have $\bar{X}(\mathcal{T})=2$ for all $\mathcal{T}\in\mathcal{T}^*(X)$ such that $f\notin\mathcal{T}$.

Let us now consider $\mathcal{P}\in\mathcal{P}^*(X)$ such that $f\in\mathcal{P}$. As $X(f)=1$, we have $X(e)=0$ for all $e\in\mathcal{P}\setminus\{f\}$. Thus, $\mathcal{P}\setminus\{f\}\subset E_0(X)$. This implies $\bar{X}(e)=Y(e)=0$ for all $e\in\mathcal{P}\setminus\{f\}$. Consequently, $\bar{X}(\mathcal{P})=1$. Let $\mathcal{T}\in\mathcal{T}^*(X)$ such that $f\in\mathcal{T}$. W.l.o.g., we can suppose that $f\notin\mathcal{P}$, where $\mathcal{P}$ is the unique path in $G(\mathcal{T})$ from n_1 to n_2. It is easy to see that $X(\mathcal{P})=1$ and $X(e)=0$ for all $e\in\mathcal{T}\setminus(\mathcal{P}\cup\{f\})$. Since $X(\mathcal{P})=1$ and $f\notin\mathcal{P}$, we have $\bar{X}(\mathcal{P})=Y(\mathcal{P})=1$ and $\bar{X}(e)=Y(e)=0$ for all $e\in\mathcal{T}\setminus(\mathcal{P}\cup\{f\})$. Thus, $\bar{X}(\mathcal{T})=2$. We conclude that $\bar{X}$ is a solution of $S(X)$, which is a contradiction, since $\bar{X}\neq X$. □

Let $G=(V,E)$ be a graph and $A=\{n_1,n_2,n_3\}$ be a terminal set. A graph G^* is a *minor* of G if we can obtain it from G by deleting and contracting edges (and deleting isolated nodes). The terminal nodes of G^*, A^* will be defined as follows: after deleting an edge, the terminals remain the same, and after contracting an edge $e=(uv)$, the new node is a terminal if and only if at least one of u, v was a terminal. The pair (G^*,A^*) will be called a minor of (G,A).

THEOREM 2.3 (see [3]). *Every minor of a nice pair is nice.*

DEFINITION 2.4 (see [3]). *The pair (G, A) will be called a minimal pair if it is not nice but all its proper minors are nice.*

We have the following lemma; its proof is analogous to that of Lemma 2.2 and is omitted.

LEMMA 2.5. *Let $G = (V, E)$ be a graph, $A = \{n_1, n_2, n_3\}$ be a terminal set, and X be an extreme point of LP(G, A). Suppose that there exists $e \in E$ such that $X(e) = 0$. Let (G^*, A^*) be a minor pair of (G, A) obtained from it by contracting e. Then X^*, the restriction of X on the graph G^*, is an extreme point of LP(G^*, A^*).*

An immediate consequence of Lemmas 2.2 and 2.5 is the following.

LEMMA 2.6. *Let (G, A) be a minimal pair and X be a noninteger extreme point of LP(G, A). Then $0 < X(e) < 1$ for all $e \in E$.*

Cunningham [3] gave the two pairs, which are minimal, shown in Figure 2.1.

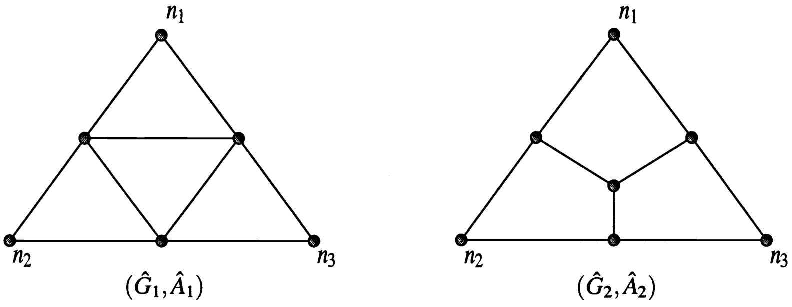


FIG. 2.1. *Minimal pairs.*

The solution $X(e) = \frac{1}{2}$ for all e is an extreme point of $\mathrm{LP}(\widehat{G}_1, \widehat{A}_1)$. The solution $X(e) = \frac{1}{2}$ if $e \in \delta(\{n_1, n_2, n_3\})$ and $X(e) = \frac{1}{4}$ elsewhere is an extreme point of $\mathrm{LP}(\widehat{G}_2, \widehat{A}_2)$

Given a graph $G = (V, E)$ and a set $A = \{n_1, n_2, n_3\} \subset V$ of terminals nodes, we denote by $\mathrm{Q}(G, A)$ the polytope defined by

$$\mathrm{Q}(G, A) = \{x \in \mathbb{R}^{|E|} \mid 0 \leq x \leq 1; x \text{ satisfies (3.1) and (3.2)}\}.$$

It is clear that $\mathrm{Q}(G, A) \subset \mathrm{LP}(G, A)$ and, by Lemma 2.1, every extreme point of $\mathrm{LP}(G, A)$ is also an extreme point of $\mathrm{Q}(G, A)$. We can also see that if X is an extreme point of $\mathrm{Q}(G, A)$ such that $X(e) < 1$ for all $e \in E$, then X is also an extreme point of $\mathrm{LP}(G, A)$.

LEMMA 2.7. *Let (G, A) be a minimal pair and X be a nonintegral extreme point of Q(G, A). Then $X(e) < 1$ for all $e \in E$ and consequently, X is an extreme point of LP(G, A).*

Proof. Let X be a nonintegral extreme point of $\mathrm{Q}(G, A)$. Assume that there exists a set of edges $F \subset E$ such that $X(e) = 1$ for all $e \in F$ and $X(e) < 1$ for all $e \in E \setminus F$. Then X^*, the restriction of X to the graph $G^* = (V, E \setminus F)$, is an extreme point of $\mathrm{Q}(G^*, A)$ and consequently, it is also an extreme point of $\mathrm{LP}(G, A)$. But this contradicts the minimality of (G, A). □

An immediate consequence of Lemma 2.7 is the following.

LEMMA 2.8. *Let (G, A) be a minimal pair and X be a noninteger extreme point of LP(G, A). Let f be an edge of E such that $0 < X(e) < 1$. Consider the following* *solution:*

$$\bar{X}(e) = \begin{cases} X(e) & \textit{if } e \in E \setminus \{f\}, \\ 1 & \textit{if } e = f. \end{cases}$$

Then $\bar{X}$ can be written as a convex combination of integral extreme points of $Q(G, A)$.

3. The main result. In this section we give necessary and sufficient conditions for a pair (G, A) to be nice.

Theorem 3.1. *A pair (G, A) is nice if and only if it contains neither $(\widehat{G}_1, \widehat{A}_1)$ nor $(\widehat{G}_1, \widehat{A}_2)$ as a minor.*

Note that by Theorem 2.3 we need only to prove that a minimal pair is either $(\widehat{G}_1, \widehat{A}_1)$ or $(\widehat{G}_1, \widehat{A}_1)$. To prove this theorem we start from a minimal pair (G, A), where $G = (V, E)$, and a noninteger extreme point X of $\mathrm{LP}(G, A)$. We are going to establish properties of (G, A) and X.

As a consequence of Lemma 2.6, we have $E_0(X) = \emptyset$ and $E(\{n_1, n_2, n_3\}) = \emptyset$.

Proposition 3.2. *The graph $G(V \setminus A)$ is connected.*

Proof. Assume that $G(V \setminus A)$ is not connected. Let $U \subset V \setminus A$ be such that $G(U)$ is a connected component of $G(V \setminus A)$. Since $E(\{n_1, n_2, n_3\}) = \emptyset$, it is easy to see that if $L \subset E$ is an A-path or an A-tree, then either $L \subset E(U \cup A)$ or $L \subset E(V \setminus U)$. Let X_1 (resp., X_2) be the restriction of X on $E(U \cup A)$ (resp., $E(V \setminus U)$). Obviously, $X_1 \in \mathrm{LP}(G(U \cup A), A)$ and $X_2 \in \mathrm{LP}(G(V \setminus U), A)$. By hypothesis, $(G(U \cup A), A)$ and $(G(V \setminus U), A)$ are nice. Thus, there exists an integer extreme point $Y_1 \in \mathrm{LP}(G(U \cup A), A)$ (resp., $Y_2 \in \mathrm{LP}(G(V \setminus U), A)$) such that every tight constraint for X_1 (resp., X_2) is also tight for Y_1 (resp., Y_2). The solution Y defined by $Y(e) = Y_1(e)$ if $e \in E(U \cup A)$ and $Y(e) = Y_2(e)$ elsewhere is a solution of $S(X)$, which is a contradiction since $Y \neq X$. □

Proposition 3.3. *Each variable $x(e)$ has a nonzero coefficient in at least two equations of $S(X)$.*

Proof. It is clear that each variable $x(e)$ must have a nonzero coefficient in at least one of the equations of $S(X)$. Assume that there exists an edge $e_0 \in E$ such that $x(e_0)$ has a nonzero coefficient in exactly one equation of $S(X)$, say, $x(L) = p$, where $p = 1$ if L is an A-path and $p = 2$ if L is an A-tree. Let G^* be the graph obtained from G by deleting e_0 and X^* be the restriction of X on $E \setminus \{e_0\}$. Let $S^*(X)$ be the system obtained from $S(X)$ by deleting the equation $x(L) = p$. By hypothesis, (G^*, A) is nice. Since X^* is not integer, it is not an extreme point of $\mathrm{LP}(G^*, A)$. Thus, there exists an extreme point $Y \in \mathrm{LP}(G^*, A)$ such that it is a solution of $S^*(X)$. Notice that Y is integer. Now consider the following solution:

$$\bar{X}(e) = \begin{cases} Y(e) & \text{if } e \neq e_0, \\ p - Y(L \setminus \{e_0\}) & \text{if } e = e_0. \end{cases}$$

Since Y is integer $\bar{X}$ is also integer. Thus, $\bar{X} \neq X$. Moreover, $\bar{X}$ is a solution of $S(X)$, which is impossible. □

Let $\mathcal{P}_{n_i n_j} = \{\mathcal{P} \in \mathcal{P}(X);\ \mathcal{P} \text{ is a path from } n_i \text{ to } n_j\}$, $i, j \in \{1, 2, 3\}$, $i \neq j$. Note that $\mathcal{P}_{n_i n_j} = \mathcal{P}_{n_j n_i}$. Let us consider the four subsets V_1, V_2, V_3, $V_0 = V \setminus (A \cup V_1 \cup V_2 \cup V_3)$ defined by

$$V_1 = \{v \in V \setminus A \mid \exists\, \mathcal{P} \in \mathcal{P}_{n_1 n_2} \text{ such that } v \in V(\mathcal{P})\},$$

$$V_2 = \{v \in V \setminus A \mid \exists\, \mathcal{P} \in \mathcal{P}_{n_2 n_3} \text{ such that } v \in V(\mathcal{P})\},$$

$$V_3 = \{v \in V \setminus A \mid \exists\, \mathcal{P} \in \mathcal{P}_{n_1n_3} \text{ such that } v \in V(\mathcal{P})\}.$$

PROPOSITION 3.4. *$\{V_0, V_1, V_2, V_3\}$ is a partition of $V \setminus A$.*

Proof. We need only to prove that $V_i \cap V_j = \emptyset$ for all $i, j \in \{1, 2, 3\}$ such that $i \neq j$. If, say, $V_1 \cap V_2 \neq \emptyset$, let v be a node of $v \in V_1 \cap V_2$. Therefore, there are $\mathcal{P}_1 \in \mathcal{P}_{n_1n_2}$ and $\mathcal{P}_2 \in \mathcal{P}_{n_2n_3}$ such that $v \in V(\mathcal{P}_1) \cap V(\mathcal{P}_2)$. Let $\mathcal{T} \subset \mathcal{P}_1 \cup \mathcal{P}_2$ be an A-tree (obviously, such A-tree exists). If there is some edge $e \in \mathcal{P}_1 \cap \mathcal{P}_2$, then $2 \leq X(\mathcal{T}) \leq X(\mathcal{P}_1 \cup \mathcal{P}_2) \leq X(\mathcal{P}_1) + (\mathcal{P}_2) - X(e) = 2 - X(e) < 2$, a contradiction.

Now suppose that $\mathcal{P}_1 \cap \mathcal{P}_2 = \emptyset$. We have $2 \leq X(\mathcal{T}) < X(\mathcal{P}_1 \cup \mathcal{P}_2) \leq X(\mathcal{P}_1) + X(\mathcal{P}_2) = 2$, a contradiction. □

PROPOSITION 3.5. *$\mathcal{P}^*(X) \neq \emptyset$.*

Proof. Assume the contrary. Let $\bar{X}$ be the solution defined as

$$\bar{X}(e) = \begin{cases} 2 & \text{if } e \in \delta(n_1), \\ 0 & \text{otherwise.} \end{cases}$$

Let $\mathcal{T} \in \mathcal{T}^*(X)$. Since $|T \cap \delta(n_1)| = 1$, we have $\bar{X}(\mathcal{T}) = 2$. Thus, $\bar{X}$ is also a solution of $S(X)$. This contradicts the fact that X is the unique solution of $S(X)$. □

PROPOSITION 3.6. *$\mathcal{T}^*(X) \neq \emptyset$.*

Proof. Suppose on the contrary that $\mathcal{T}^*(X) = \emptyset$. Let $\bar{X}$ be the solution given by

$$\bar{X}(e) = \begin{cases} 1 & \text{if } e \in \delta(n_1) \cap \delta(V_3), \\ 1 & \text{if } e \in \delta(n_2) \cap \delta(V_1), \\ 1 & \text{if } e \in \delta(n_3) \cap \delta(V_2), \\ 0 & \text{otherwise.} \end{cases}$$

$\bar{X}$ is also a solution of $S(X)$; this contradicts the extremality of X, since $\bar{X} \neq X$. □

PROPOSITION 3.7. *$\mathcal{P}_{n_in_j} \neq \emptyset$, i, $j \in \{1,\ 2,\ 3\}$, $i \neq j$.*

Proof. If, say, $\mathcal{P}_{n_1n_2} = \emptyset$, then let $\bar{X}$ be the following solution:

$$\bar{X}(e) = \begin{cases} 1 & \text{if } e \in \delta(n_1) \cup \delta(n_2), \\ 0 & \text{otherwise.} \end{cases}$$

If $\mathcal{P} \in \mathcal{P}(X) = \mathcal{P}_{n_1n_3} \cup \mathcal{P}_{n_2n_3}$, then $|\mathcal{P} \cap (\delta(n_1) \cup \delta(n_2))| = 1$. Thus, $\bar{X}(\mathcal{P}) = 1$. Let $\mathcal{T} \in \mathcal{T}(X)$. Since $|\mathcal{T} \cap (\delta(n_1) \cup \delta(n_2))| = 2$, we have $\bar{X}(\mathcal{T}) = 2$. Thus, $\bar{X}$ is also a solution of $S(X)$, which is a contradiction. □

Since $X(e) < 1$ for all $e \in E$, Proposition 3.7 implies that $V_i \neq \emptyset$, $i = 1, 2, 3$. (If, for instance, $V_1 = \emptyset$, then $e_0 = (n_1n_2) \in E$ and $X(e_0) = 1$, which is impossible.)

PROPOSITION 3.8. *If $v \in V \setminus A$, then $|\delta(v)| \geq 3$.*

Proof. It is easy to see that $|\delta(v)| \geq 2$ for all $v \in V \setminus A$. Assume that there exists some $u \in V \setminus A$ such that $\delta(u) = \{e_1, e_2\}$. Let $\bar{X}$ be the solution given by

$$\bar{X}(e) = \begin{cases} X(e) & \text{if } e \in E \setminus \delta(u), \\ X(e_1) + X(e_2) & \text{if } e = e_1, \\ 0 & \text{if } e = e_2. \end{cases}$$

$\bar{X}$ is also a solution of $S(X)$, which is a contradiction. □

PROPOSITION 3.9. *$\delta(V_i) \setminus (\delta(n_i) \cup \delta(n_{i+1})) \neq \emptyset$, $i = 1, 2, 3$. (For convenience, $n_4 = n_1$.)*

Proof. The proof is an immediate consequence of Proposition 3.2. □

PROPOSITION 3.10. *$\delta(V_i) \cap \delta(n_{i+2}) = \emptyset$, $i = 1, 2, 3$* (mod 3).

Proof. Assume, for instance, that there exists some node $v_1 \in V_1$ such that $f = (v_1 n_3) \in E$. By Lemma 2.6, $X(f) < 1$. Let $\mathcal{P} \in \mathcal{P}_{n_1 n_2}$ such that $v_1 \in V(\mathcal{P})$ and $\mathcal{T} = \mathcal{P} \cup \{e\}$. $\mathcal{T}$ is an A-tree such that $X(\mathcal{T}) = X(\mathcal{P}) + X(e) = 1 + X(f) < 2$, a contradiction. □

PROPOSITION 3.11. *If P is a path from n_i to some node $v \in V_{i+1}$ (for convenience $V_4 = V_1$), $i \in \{1, 2, 3\}$, then $X(P) \geq 1$.*

Proof. Consider the case $i = 1$. (The proofs for others cases are similar.) W.l.o.g., we can suppose that $V(P) \cap V_2 = \{v\}$. It is clear that $X(P) \geq 1$ if $V(P)$ contains n_2 or n_3. Suppose that n_2, $n_3 \notin V(P)$. Let $\mathcal{P} \in \mathcal{P}_{n_2 n_3}$ such that $v \in V(\mathcal{P})$ and $\mathcal{T} = P \cup \mathcal{P}$. Since $\mathcal{T}$ is an A-tree, we have $X(\mathcal{T}) = X(P) + X(\mathcal{P}) \geq 2$. Thus, $X(P) \geq 1$. □

PROPOSITION 3.12. *If P is a path from n_i to $v_{i+1} \in V_{i+1}$ such that $V(P) \cap V_i \neq \emptyset \neq V(P) \cap V_{i+2}$, then $X(P) > 1$.*

Proof. We consider the case $i = 1$. By Proposition 3.11, $X(P) \geq 1$. Suppose that $X(P) = 1$. Let $v_1 \in V(P) \cap V_1$ and $v_3 \in V(P) \cap V_3$. Suppose, w.l.o.g., that $v_1 \notin V(P_{v_2 v_3})$ and $V(P_{v_2 v_3}) \cap V_3 = \{v_3\}$, where $P_{v_2 v_3}$ is a part of P from v_2 to v_3. Since $v_3 \in V_3$ there must exist $\mathcal{P} \in \mathcal{P}_{n_1 n_3}$ such that $v_3 \in V(\mathcal{P}_1)$. Note that $\mathcal{P} \cap P_{v_2 v_3} = \emptyset$. Let $\mathcal{P}_{v_3 n_1}$ be a part of $\mathcal{P}$ from v_3 to n_1. Since $\mathcal{P}_{v_3 n_1} \cup P_{v_2 v_3}$ is path from n_1 to v_2, Proposition 3.11 gives $X(\mathcal{P}_{v_3 n_1} \cup P_{v_2 v_3}) = X(\mathcal{P}_{v_3 n_1}) + X(P_{v_2 v_3}) \geq 1$. Thus,

$$X(P_{v_2 v_3}) \geq 1 - X(\mathcal{P}_{v_3 n_1}) = X(\mathcal{P} \setminus \mathcal{P}_{v_3 n_1}). \tag{3.1}$$

Now let $F = (P \setminus P_{v_2 v_3}) \cup (\mathcal{P} \setminus \mathcal{P}_{v_3 n_1})$. Since $G(F)$ is connected and $\{n_1, n_3\} \subset V(F)$, we have $1 \leq X(F) \leq X(P) - X(P_{v_2 v_3}) + X(\mathcal{P} \setminus \mathcal{P}_{v_3 n_1})$. From (3.1), we obtain $1 \leq X(F) \leq X(P) = 1$. Thus, $X(F) = 1$ and hence F is a path from n_1 to n_3. Since $v_1 \in V(F)$, then by definition $v_1 \in V_3$. Thus, $v_1 \in V_1 \cap V_3$, contradicting Proposition 3.4. □

PROPOSITION 3.13. *Let $\mathcal{T} \in \mathcal{T}^*(X)$ and $v \in V$ be the unique node such that $|\delta(v) \cap \mathcal{T}| = 3$. Suppose that $v \in V_i$ for a certain $i \in \{1, 2, 3\}$. Then $\mathcal{T} = \mathcal{P} \cup P$, where $\mathcal{P} \in \mathcal{P}_{n_i n_{i+1}}$ and P is a path from v to n_{i+2} such that $V(P) \cap V_{i+1} = \emptyset$ or $V(P) \cap V_{i+2} = \emptyset$.*

Proof. We consider the case $i = 1$. Let $\mathcal{P}$ be the unique path in $G(\mathcal{T})$ from n_1 to n_2. It is clear that $v \in \mathcal{P}$. Let $P = \mathcal{T} \setminus \mathcal{P}$. Since P is a path from v to n_3, then by Proposition 3.11 $X(P) \geq 1$. This implies that $X(\mathcal{P}) = 1$ and $X(P) = 1$. Consequently, $\mathcal{P} \in \mathcal{P}_{n_1 n_2}$. By Proposition 3.12 we have $V(P) \cap V_{i+1} = \emptyset$ or $V(P) \cap V_{i+2} = \emptyset$. □

PROPOSITION 3.14. *Let $\mathcal{T} \in \mathcal{T}^*(X)$. Let $v \in V$ be the unique vertex such that $|\delta(v) \cap \mathcal{T}| = 3$. Suppose that $v \in V_0$. Let P_i be the unique path from v to n_i, $i = 1, 2, 3$. Then $V(P_i) \subset V_0 \cup V_j$ for a certain $j \in \{i, i+2\}$. Moreover, if $V(P_i) \cap V_j \neq \emptyset$, then there exists some node $u \in V_j$ such that $P_i = P_i^1 \cup P_i^2$, where P_i^1 is a path from v to u such that $V(P_i^1) \subset V_0 \cup \{u\}$ and P_i^2 is a path from u to n_i such that $V(P_i^2) \subset V_j \cup \{n_i\}$.*

Proof. We are going to prove the result for $i = 1$. Let $u \in V(P_1) \setminus V_0$ and P_1^1 be the path in $G(\mathcal{T})$ from u to n_1. If $u \in V_2$, then by Proposition 3.11 we have $X(P_1^1) \geq 1$. This implies that $X(P_1) > 1$ and, consequently, $X(P_2 \cup P_3) < 1$, which is impossible since $P_2 \cup P_3$ is a path from n_2 to n_3.

Suppose now, w.l.o.g., that $u \in V_1$. We shall prove that $V(P_1^1) \subset V_1 \cup \{n_1\}$. Let $u' \in V(P_1^1) \setminus \{u, n_1\}$. Since $u \in V_1$, there must exist $\mathcal{P} \in \mathcal{P}_{n_1 n_2}$ such that $u \in V(\mathcal{P})$. If $u' \in V(\mathcal{P})$, then $u' \in V_1$. Suppose that $u' \notin V(\mathcal{P})$. Let $\mathcal{P}^*$ be the path in $G(\mathcal{P})$ from n_2 to u, and let $\mathcal{P}' = \mathcal{P}^* \cup P_1^1$. It easy to see that $X(\mathcal{P}') = 1$. Since $\mathcal{P}'$ is a path from n_1 to n_2 and $u' \in V(\mathcal{P}')$, by definition of V_1, we have $u' \in V_1$. □

Proposition 3.15. *Let $\mathcal{T} \in \mathcal{T}^*(X)$ and $v \in V(\mathcal{T})$ such that $|\delta(v) \cap \mathcal{T}| = 3$. Suppose that there are $v_i \in V_i$ and $v_j \in V_j$, $i, j \in \{1,2,3\}$, $i \neq j$, such that $f = (v_iv_j) \in \mathcal{T}$. Then $v \in \{v_i, v_j\}$.*

Proof. Suppose, on the contrary, that $v \notin \{v_i, v_j\}$. W.l.o.g., suppose that $i = 1$ and $j = 3$. Let P_s be the unique path in $G(\mathcal{T})$ from n_s to v, $s = 1, 2, 3$. We claim that $f \in P_1$. In fact, if $f \in P_2$, then by Proposition 3.11 and the assumption $v \notin \{v_1, v_3\}$ we will have $X(P_2) > 1$. This implies $X(P_1 \cup P_3) < 1$, since $\mathcal{T} = P_1 \cup P_2 \cup P_3$ and $X(\mathcal{T}) = 2$. But $P_1 \cup P_2$ is a path from n_1 to n_3, a contradiction. By the same arguments, we can proof that $f \notin P_3$. Consequently, we have $f \in P_1$.

Let $P_{n_1v_1}$ be the unique path in $G(\mathcal{T})$ from n_1 to v_1. (Note that $P_{n_1v_1} \subset P_1$.)

Suppose that $f \in P_{n_1v_1}$. (The case $f \notin P_{n_1v_1}$ is similar.) Let $\mathcal{P}_1 \in \mathcal{P}_{n_1n_2}$ such that $v_1 \in V(\mathcal{P}_1)$. Let $\mathcal{P}_{n_2v_1}$ be the path in $G(\mathcal{P}_1)$ from n_2 to v_1 and $\mathcal{P}_{v_1n_1}$ be the path in $G(\mathcal{P}_1)$ from v_1 to n_1. Let $\mathcal{T}_1 = (\mathcal{T} \setminus P_{n_1v_1}) \cup \mathcal{P}_{v_1n_1}$. We have, $2 \leq X(\mathcal{T}_1) \leq X(\mathcal{T}) - X(P_{n_1v_1}) + X(\mathcal{P}_{v_1n_1})$. Hence,

$$X(\mathcal{P}_{v_1n_1}) \geq X(P_{n_1v_1}). \tag{3.2}$$

Let $\mathcal{P}' = \mathcal{P}_{n_2v_1} \cup P_{n_1v_1}$. We have

$$1 \leq X(\mathcal{P}') \leq X(\mathcal{P}_{n_2v_1}) + X(P_{n_1v_1}) = X(\mathcal{P}_1) - X(\mathcal{P}_{v_1n_1}) + X(P_{n_1v_1}).$$

This implies that

$$X(P_{n_1v_1}) \geq X(\mathcal{P}_{v_1n_1}). \tag{3.3}$$

By (3.2) and (3.3), we have $X(\mathcal{P}_{v_1n_1}) = X(P_{n_1v_1})$. We then obtain $X(\mathcal{P}') = 1$. Since $v_3 \in V(\mathcal{P}')$ and $\mathcal{P}' \in \mathcal{P}_{n_1n_2}$, we have $v_3 \in V_1$. Thus, $v_3 \in V_1 \cap V_3$, which is impossible. □

4. Proofs of Theorem 3.1. To complete the proof of our theorem we shall distinguish two cases.

Case 1. There exists an edge $f \in [V_1, V_2] \cup [V_1, V_3] \cup [V_2, V_3]$. W.l.o.g., suppose that $f = (v_1v_3) \in [V_1, V_3]$, where $v_1 \in V_1$ and $v_3 \in V_3$.

Claim 1. *We can choose $S(X)$ such that $x(f)$ has a nonzero coefficient in exactly two equations of $S(X)$.*

Proof. Let $\mathcal{T}_1 \in \mathcal{T}^*(X)$ such that $f \in \mathcal{T}_1$. By Proposition 3.15, either $|\delta(v_1) \cap \mathcal{T}_1| = 3$ or $|\delta(v_3) \cap \mathcal{T}_1| = 3$. Suppose, w.l.o.g., that $|\delta(v_1) \cap \mathcal{T}_1| = 3$. Suppose that there is $\mathcal{T}_1' \in \mathcal{T}^*(X)$ such that $f \in \mathcal{T}_1'$ and $|\delta(v_1) \cap \mathcal{T}_1'| = 3$. By Proposition 3.13, $\mathcal{T}_1 = \mathcal{P}_1 \cup P_1$ and $\mathcal{T}_1' = \mathcal{P}_1' \cup P_1'$, where $\mathcal{P}_1$, $\mathcal{P}_1' \in \mathcal{P}_{n_1n_2}$ and $V(P_1) \cap V_2 = \emptyset = V(P_1') \cap V_2$. Since $X(\mathcal{P}_1) = X(\mathcal{P}_1') = 1$, we have $X(P_1 \setminus \{f\}) = X(P_1' \setminus \{f\})$. It is clear that if P is a path from n_3 to v_3, then $X(P) \geq X(P_1 \setminus \{f\}) = X(P_1' \setminus \{f\})$. Otherwise, $\mathcal{T} = \mathcal{P}_1 \cup (P \cup \{f\})$ would be an A-tree such that $X(\mathcal{T}) < 2$, which is impossible. Let $\mathcal{P}^* \in \mathcal{P}_{n_1n_3}$ such that $v_3 \in V(\mathcal{P}^*)$. $\mathcal{P}^* = P_1^* \cup P_2^*$, where P_1^* is a path from n_3 to v_3 and P_2^* is a path from v_3 to n_1. Clearly, $X(P_1^*) \leq X(P_1 \setminus \{f\}) = X(P_1' \setminus \{f\})$. Thus, $X(P_1^*) = X(P_1 \setminus \{f\}) = X(P_1' \setminus \{f\})$. Let $\mathcal{P}_1^* = (P_1 \setminus \{f\}) \cup P_2^*$ and $\mathcal{P}_2^* = (P_1' \setminus \{f\}) \cup P_2^*$. Thus, $\mathcal{P}_1^*$, $\mathcal{P}_2^* \in \mathcal{P}_{n_1n_3}$. Hence $x(\mathcal{T}_1') = 2$ is redundant with respect $x(\mathcal{T}_1) = 2$, $x(\mathcal{P}_1) = 1$, $x(\mathcal{P}_1') = 1$, $x(\mathcal{P}_1^*) = 1$, and $x(\mathcal{P}_2^*) = 1$. Consequently, one may assume that $x(\mathcal{T}_1') = 2$ does not belong to $S(X)$. By Proposition 3.3, there is $\mathcal{T}_2 \in \mathcal{T}^*(X)$ such that $f \in \mathcal{T}_2$ and $\mathcal{T}_2 \neq \mathcal{T}_1$. Thus, $|\delta(v_3) \cap \mathcal{T}_2| = 3$. Using the same arguments as above, we can prove that if $\mathcal{T} \in \mathcal{T}^*(X)$ such that $|\delta(v_3) \cap \mathcal{T}| = 3$ and $\mathcal{T} \neq \mathcal{T}_2$, then $S(X)$ can be chosen such that $\mathcal{T} \notin S(X)$, and this completes the proof of our claim. □

Claim 2. *$X(e) \in \{\alpha, 1-\alpha\}$ for a certain $\alpha \in]0,1[$ for all $e \in E$.*

Proof. Let $\bar{X}$ be the solution given by

$$\bar{X}(e) = \begin{cases} X(e) & \text{if } e \in E \setminus \{f\}, \\ 1 & \text{if } e = f. \end{cases}$$

Since $\bar{X}$ is not an integer (otherwise, X would be a noninteger extreme point with one fractional component, which is impossible), by Lemma 2.6, $\bar{X}$ is not an extreme point of LP(G, A). By Lemma 2.8, there must exist $t \geq 2$ integral extreme points of Q(G, A) and t scalars $0 < \lambda_i < 1$, $i = 1, \dots, t$, such that

$$\bar{X} = \sum_{i=1}^{t} \lambda_i Y_i \quad \text{and} \quad \sum_{i=1}^{t} \lambda_i = 1.$$

By Claim 1, $S(X)$ can be chosen such that f belongs to exactly two A-trees of $\mathcal{T}^*(X)$, namely, $\mathcal{T}_1$ and $\mathcal{T}_2$. Let $S^*(X)$ be the system obtained from $S(X)$ by removing $x(\mathcal{T}_1) = 2$ and $x(\mathcal{T}_2) = 2$. Clearly, Y_i is a solution of $S^*(X)$ and $Y_i(f) = 1$ for $i = 1, \dots, t$. Since $\bar{X}(\mathcal{T}_1) + \bar{X}(\mathcal{T}_2) = 4 + 2 - 2X(f) < 6$, there must exist a certain $i \in \{1, \dots, t\}$ such that $Y_i(\mathcal{T}_1) + Y_i(\mathcal{T}_2) < 6$. W.l.o.g., suppose that $i = 1$. Since Y_1 is integer, we have $Y_1(\mathcal{T}_1) + Y_1(\mathcal{T}_2) \leq 5$. If $Y_1(\mathcal{T}_1) + Y_1(\mathcal{T}_2) = 4$, then we would have $Y_1(\mathcal{T}_1) = Y_1(\mathcal{T}_2) = 2$. Thus, Y_1 is also a solution of $S(X)$, a contradiction. Consequently, $Y_1(\mathcal{T}_1) + Y_1(\mathcal{T}_2) = 5$. W.l.o.g., we can suppose that $Y_1(\mathcal{T}_1) = 2$ and $Y_1(\mathcal{T}_2) = 3$. Since $\bar{X}(\mathcal{T}_2) = 2 + 1 - X(f) < 3$, there must exist a certain $j \in \{1, \dots, t\}$ such that $Y_j(\mathcal{T}_2) < 3$. Since Y_j is integer, we have $Y_j(\mathcal{T}_2) = 2$. Thus, $j \neq 1$. W.l.o.g., suppose that $j = 2$. Obviously, $Y_2(\mathcal{T}_1) \geq 3$. Define

$$\alpha = \frac{Y_2(\mathcal{T}_1) - 2}{Y_2(\mathcal{T}_1) - 1} \quad \text{and} \quad Z^* = \alpha Y_1 + (1-\alpha) Y_2.$$

It is easy to check that $\alpha \in]0,1[$ and $Z^*(\mathcal{T}_1) = Z^*(\mathcal{T}_2)$. Thus, $Z^*(\mathcal{T}_1 \setminus \{f\}) = Z^*(\mathcal{T}_2 \setminus \{f\})$. Let Z be the following solution:

$$Z(e) = \begin{cases} Z^*(e) & \text{if } e \in E \setminus \{f\}, \\ 2 - Z^*(\mathcal{T}_1 \setminus \{f\}) & \text{if } e = f. \end{cases}$$

Clearly Z is a solution of $S(X)$ (we need only to check that $Z(\mathcal{T}_1) = Z(\mathcal{T}_1) = 2$). This implies that $X = Z$. Since $0 < X(e) < 1$ for all $e \in E$, we have $Y_1(e)Y_2(e) = 0$ for all $e \in E \setminus \{f\}$. (If, for instance, there is $e_0 \neq f$ such that $Y_1(e_0) = Y_2(e_0) = 1$, then we would have $X(e_0) = 1$.) Thus, $X(e) \in \{\alpha, 1-\alpha\}$ for all $e \in E \setminus \{f\}$. Furthermore, we have $X(f) = Z(f) = 2 - Z^*(\mathcal{T}_2 \setminus \{f\}) = 2 - (\alpha Y_1(\mathcal{T}_2 \setminus \{f\}) + (1-\alpha) Y_2(\mathcal{T}_2 \setminus \{f\}) = 2 - (2\alpha + (1-\alpha)) = 1 - \alpha$. □

Claim 3. *$|\mathcal{P}| = 2$ for all $\mathcal{P} \in \mathcal{P}(X)$ and $|\mathcal{T}| = 4$ for all $\mathcal{T} \in \mathcal{T}(X)$.*

Proof. Suppose that there is $\mathcal{P} \in \mathcal{P}(X)$ such that $|\mathcal{P}| \geq 3$. Since $X(e) \in \{\alpha, 1-\alpha\}$ for all $e \in E$, there are at least two edges e_1, $e_2 \in \mathcal{P}$ such that $X(e_1) = X(e_2)$. Suppose, w.l.o.g., that $X(e_1) = X(e_2) = \alpha$. Thus, $Y_1(e_1) = Y_1(e_2) = 1$. (Y_1 and Y_2 are defined in Claim 2.) Consequently, $Y_1(\mathcal{P}) \geq 2$. This contradicts the fact that Y_1 is a solution of $S^*(X)$. By the same approach we can prove that $|\mathcal{T}| = 4$. □

An immediate consequence of the last claim is $E(V_i) = \emptyset$ for all $i \in \{1, 2, 3\}$.

Claim 4. *$V \setminus A = V_1 \cup V_2 \cup V_3$.*

Proof. Suppose that $V_0 = V \setminus (A \cup V_1 \cup V_2 \cup V_3) \neq \emptyset$. Let $\mathcal{T}_0 \in \mathcal{T}^*(X)$ such that $E(\mathcal{T}_0) \cap \delta(V_0) \neq \emptyset$. Let v_0 be the only node such that $|\delta(v_0) \cap \mathcal{T}_0| = 3$. By Claim 3

$E(\mathcal{T}_0) = \{e_1 = (v_0n_i), e_2 = (v_0n_j), e_3 = (v_0u_0), e_4 = (u_0n_l)\}$, where $\{i, j, l\} = \{1, 2, 3\}$ and $u_0 \in V \setminus (A \cup \{v_0\})$. Suppose, w.l.o.g., that $i = 2$, $j = 3$ and consequently $l = 1$. First, we shall prove that $v_0 \in V_1 \cup V_2 \cup V_3$. If $v_0 \notin V_2$, then we should have $X(e_1) + X(e_2) > 1$. This implies that $X(e_1) = X(e_2)$ (otherwise, $X(e_1) + X(e_2) = \alpha + 1 - \alpha = 1$). Suppose, w.l.o.g., that $X(e_1) = X(e_2) = \alpha$. Since $X(e) = \alpha Y_1(e) + (1 - \alpha)Y_2(e)$ for all $e \in E \setminus \{f\}$ (Y_1 and Y_2 are the same points defined in Claim 2), we have $Y_2(e_1) = Y_2(e_2) = 0$. This contradicts that $Y_2 \in \mathrm{LP}(G, A)$. Consequently, $v_0 \in V_1 \cup V_2 \cup V_3$. Since $E(\mathcal{T}_0) \cap \delta(V_0) \neq \emptyset$, we have $u_0 \in V_0$.

Now consider the solution $\bar{X}$ defined as

$$\bar{X}(e) = \begin{cases} X(e) & \text{if } e \in E \setminus ([\{u_0\}, V_2] \cup \{e_4\}), \\ 1 & \text{if } e = (u_0n_1), \\ 0 & \text{if } e \in [\{u_0\}, V_2]. \end{cases}$$

Let $\mathcal{T} \in \mathcal{T}(X)$. If $\mathcal{T} \cap ([\{u_0\}, V_2] \cup \{(u_0n_1)\}) = \emptyset$, then $\bar{X}(\mathcal{T}) = X(\mathcal{T}) = 2$. Suppose that $\mathcal{T} \cap ([\{u_0\}, V_2] \cup \{(u_0n_1)\}) \neq \emptyset$. Let v be the only node such that $|\delta(v) \cap \mathcal{T}| = 3$. Since $v \in V_1 \cup V_2 \cup V_3$, $\mathcal{T} \cap ([\{u_0\}, V_2] \cup \{(u_0n_1)\}) \neq \emptyset$, and $|\mathcal{T}| = 4$, Proposition 3.13 implies that $\mathcal{T} = \{(vn_2), (vn_3), (vu_0), (u_0n_1) \in \mathcal{T}$ and $X(vu_0) + X(u_0n_1) = 1$. Thus, $\bar{X}(\mathcal{T}) = X(\mathcal{T}) = 2$. Consequently, $\bar{X}$ is also a solution of $S(X)$, which is impossible. □

CLAIM 5. $[V_1, V_2] \neq \emptyset \neq [V_2, V_3]$.

Proof. As $V_0 = \emptyset$, by Proposition 3.2 we have $[V_1, V_2] \cup [V_2, V_3] \neq \emptyset$. Suppose, w.l.o.g., that $[V_2, V_3] \neq \emptyset$. If $[V_1, V_2] = \emptyset$, then the solution defined by

$$\bar{X}(e) = \begin{cases} 1 & \text{if } e \in \delta(n_3) \cup [V_2, V_3] \cup [\{n_2\}, V_1], \\ 0 & \text{otherwise,} \end{cases}$$

is also a solution of $S(X)$, a contradiction. □

CLAIM 6. *$G(V_1 \cup V_2 \cup V_3)$ is a Hamilton cycle.*

Proof. Let $L = |E(V_1 \cup V_2 \cup V_3)|$. By Claim 3, $|E| = 2(|V_1| + |V_2| + |V_3|) + L$. For every $v \in V_1 \cup V_2 \cup V_3$, $P = \delta(v) \cap \delta(\{n_1, n_2, n_3\})$ is an A-path such that $X(P) = 1$. Every $e \in E(V_1 \cup V_2 \cup V_3)$ belongs to exactly two A-trees of $\mathcal{T}^*(X)$. Thus, by using Claims 3 and 4, $|\mathcal{T}^*(X)| = 2L$. It is not hard to see that we can choose $S(X)$ such that $\mathcal{P}^*(X) = \mathcal{P}(X)$. Since X is the unique solution of $S(X)$, we then have $|E| = |\mathcal{P}^*(X)| + |\mathcal{T}^*(X)| = 2(|V_1| + |V_2| + |V_3|) + L$. This implies that $L = |V_1| + |V_2| + |V_3|$.

Now we are going to show that $|\delta(v) \cap E(V_1 \cup V_2 \cup V_3)| = 2$ for all $v \in V_1 \cup V_2 \cup V_3$. First, let us prove that $|\delta(v) \cap E(V_1 \cup V_2 \cup V_3)| \geq 2$. Since $G(V \setminus A) = G(V_1 \cup V_2 \cup V_3)$ is connected, $|\delta(v) \cap E(V_1 \cup V_2 \cup V_3)| \geq 1$. Suppose that there exists $u \in V_1 \cup V_2 \cup V_3$ such that $\delta(u) \cap E(V_1 \cup V_2 \cup V_3) = \{(uv)\}$, where $v \in (V_1 \cup V_2 \cup V_3) \setminus \{u\}$. W.l.o.g., suppose that $u \in V_2$ and $v \in V_3$. Let us consider the graph G' obtained from G by deleting the set of edges $\{(un_2), (un_3), (uv)\}$. Let X' be the restriction of X to G'. By hypothesis, (G', A) is a nice pair. So there must exist an integral extreme point $Y' \in \mathrm{LP}(G', A)$ such that every tight constraint for X' is also tight for Y'. Let Y be the solution given by

$$Y(e) = \begin{cases} Y'(e) & \text{if } e \in E \setminus \delta(u), \\ Y'(vn_3) & \text{if } e \in \{(un_3), (uv)\}, \\ Y'(vn_1) & \text{if } e = (un_2). \end{cases}$$

Y is also a solution of $S(X)$, a contradiction.

Now suppose that there exists $s \in V_1 \cup V_2 \cup V_3 V_3$ such that $|\delta(v) \cap E(V_1 \cup V_2 \cup V_3)| \geq 3$. We have

$$\begin{aligned} 2L = \sum_{v \in (V_1 \cup V_2 \cup V_3) \setminus \{s\}} & |[\{v\}, V_1 \cup V_2 \cup V_3]| + |[\{s\}, V_1 \cup V_2 \cup V_3]| \\ & \geq 2(L-1) + 3 = 2L + 1, \end{aligned}$$

a contradiction. We have proved that $|\delta(v) \cap E(V_1 \cup V_2 \cup V_3)| = 2$ for all $v \in V_1 \cup V_2 \cup V_3$. Since $G(V_1 \cup V_2 \cap V_3)$ is connected, we then obtain that $G(V_1 \cup V_2 \cup V_3)$ is a Hamilton cycle. □

Consequently, $(\widehat{G}_1, \widehat{A}_1)$ is a minor of (G, A). Thus, $(G, A) = (\widehat{G}_1, \widehat{A}_1)$ since (G, A) is minimal.

Case 2. $[V_1, V_2] \cup [V_1, V_3] \cup [V_2, V_3] = \emptyset$.

Since $G(V \setminus A)$ is connected (Proposition 3.2), we have $V_0 = V \setminus (A \cup V_1 \cup V_2 \cup V_3) \neq \emptyset$. If $G(V_0)$ is connected, then $(\widehat{G}_2, \widehat{A}_2)$ is a minor of (G, A). Thus, $(G, A) = (\widehat{G}_2, \widehat{A}_2)$.

Suppose that $G(V_0)$ is not connected. Since $G(V \setminus A)$ is connected, there must exist $V'_k \subseteq V_k$, $V'_s \subseteq V_s$, $\{k, s\} \subset \{1, 2, 3\}$ and $k \neq s$, and $V_0^1 \subset V_0$ such that V'_k (resp., V'_s) induces a connected component of $G(V_k)$ (resp., $G(V_s)$) and for all $v \in V_0^1$ there is a path $\mathcal{P}$ from some node u_k of V'_k to some node u_s of V'_s such that $v \in V(\mathcal{P})$ and $V(\mathcal{P}) \setminus \{u_k, u_s\} \subset V_0^1$. Suppose, w.l.o.g., that $k = 1$ and $s = 3$.

Let V'_2 be a subset of V_2 such that $G(V'_2)$ is a connected component of $G(V_2)$. If there is a path $\mathcal{P}$ from some node $v_0 \in V_0^1$ to some node $v_2 \in V'_2$ such that $V(\mathcal{P}) \cap (V'_1 \cup V'_3) = \emptyset$, then $(\widehat{G}_2, \widehat{A}_2)$ is a minor of (G, A). Thus, $(G, A) = (\widehat{G}_2, \widehat{A}_2)$. If there exist a path $\mathcal{P}_1$ from some node of V'_2 to some node of V'_1 and a path $\mathcal{P}_2$ from some node of V'_2 to some node of V'_3 such that $V(\mathcal{P}_1) \cap V'_3 = \emptyset = V(\mathcal{P}_2) \cap V'_1 = \emptyset$, then $(\widehat{G}_1, \widehat{A}_1)$ is a minor of (G, A). And hence $(G, A) = (\widehat{G}_1, \widehat{A}_1)$.

Suppose now, w.l.o.g., that if $\mathcal{P}$ is a path from some node of V'_2 to some node of V'_3, then $V(\mathcal{P}) \cap V'_1 \neq \emptyset$. Let V_0^2 be a set of nodes $v \in V_0$ such that there exists a path $\mathcal{P}$ from some node of V'_1 to some node of V'_2 and $v \in V(\mathcal{P})$.

Let $f_1 = (v_0^1 v'_1)$, where $v_0^1 \in V_0^1$ and $v'_1 \in V'_1$. Since $v'_1 \in V_1$, there exists a path $\mathcal{P}_0 \in \mathcal{P}_{n_1 n_2}$ such that $v'_1 \in V(\mathcal{P}_0)$. Let U_1 (resp., U_2) be the subset of nodes $v \in V'_1 \setminus V(\mathcal{P}_0)$ such that there exists a path $\mathcal{P}$ from v to some node of V'_3 (resp., V'_2) and $V(\mathcal{P}) \cap V(\mathcal{P}_0) = \emptyset$. Let $U_3 = V'_1 \setminus (V(\mathcal{P}_0) \cup U_1 \cup U_2)$. If $U_1 \cap U_2 \neq \emptyset$, then $(\widehat{G}_2, \widehat{A}_2)$ is a minor of (G, A). By minimality of (G, A), $(G, A) = (\widehat{G}_2, \widehat{A}_2)$.

So we can assume from now on that $U_i \cap U_j = \emptyset$ and $[U_i, U_j] = \emptyset$ for all $i, j \in \{1, 2, 3\}$, $i \neq j$. Before ending the second part of our proof we need to establish the three following claims.

Claim 7. *We can choose $S(X)$ such that if $\mathcal{P} \in \mathcal{P}_{n_1 n_2}$, then $V(\mathcal{P}) \cap V'_1 \subset V(\mathcal{P}_0) \cup U_k$ for a certain subscript $k \in \{1, 2, 3\}$.*

Proof. Consider a pair $u, \bar{u}$ of $V(\mathcal{P}) \cap V(\mathcal{P}_0)$ such that $V(\mathcal{P}_0) \cap V(P_u) = \{u, \bar{u}\}$ and $V(P_u) \setminus \{u, \bar{u}\} \neq \emptyset$, where P_u is the unique path in $G(\mathcal{P})$ from u to $\bar{u}$. Since $U_i \cap U_j = \emptyset$ and $[U_i, U_j] = \emptyset$ for all $i, j \in \{1, 2, 3\}$, $i \neq j$, we must have $V(P_u) \setminus \{u, \bar{u}\} \subset U_k$ for a certain subscript $k \in \{1, 2, 3\}$. Let $\mathcal{P}_u = (\mathcal{P}_0 \setminus P) \cup P_u$, where P is the unique path in $G(\mathcal{P}_0)$ from u to $\bar{u}$. It is easy to see that $\mathcal{P}_u \in \mathcal{P}_{n_1 n_2}$ and $V(\mathcal{P}_u) \cap V'_1 \subset V(\mathcal{P}_0) \cup U_k$. The equation $x(\mathcal{P}) = 1$ can be obtained from $x(\mathcal{P}_0) = 1$ and $x(\mathcal{P}_u) = 1$ for every pair $(u, \bar{u})$ described as above. □

Claim 8. *We can choose $S(X)$ such that if $\mathcal{T} \in \mathcal{T}^*(X)$ and $V(\mathcal{T}) \cap V'_1 \neq \emptyset$, then $V(\mathcal{T}) \cap V'_1 \subset V(\mathcal{P}_0) \cup U_k$ for a certain subscript $k \in \{1, 2, 3\}$.*

Proof. Let $\mathcal{T} \in \mathcal{T}^*(X)$ and v be the only node such that $|\delta(v) \cap \mathcal{T}| = 3$. If $v \in V'_1$, then by Proposition 3.13, $\mathcal{T} = \mathcal{P} \cup P$, where $\mathcal{P} \in \mathcal{P}_{n_1 n_2}$ and P is the unique path in

$G(\mathcal{T})$ from v to n_3. First, suppose that $v \in V(\mathcal{P}_0)$. Let $\mathcal{T}' = \mathcal{P}_0 \cup P$. Clearly, we have $X(\mathcal{T}') = 2$. Thus, the equation $x(\mathcal{T}) = 2$ can be obtained from $x(\mathcal{T}') = 2$, $x(\mathcal{P}_0) = 1$ and $x(\mathcal{P}) = 1$. Suppose now that $v \in U_k$ for some $k \in \{1, 2, 3\}$. Let $\mathcal{P}_k$ be the unique path in $G(\mathcal{P} \cup \mathcal{P}_0)$ from n_1 to n_2 such that $V(\mathcal{P}) \cap U_k \subset V(\mathcal{P}_k)$, and $\mathcal{T}^* = \mathcal{P}_k \cup P$. It is easy to see that $X(\mathcal{T}^*) = 2$ and $X(\mathcal{P}_k) = 1$. Thus, the equation $x(\mathcal{T}) = 2$ can be obtained from $x(\mathcal{T}^*) = 2$, $x(\mathcal{P}_k) = 1$ and $x(\mathcal{P}) = 1$. In both cases, by Claim 7, we can choose $S(X)$ such that $V(\mathcal{P}) \cap V_1' \subset V(\mathcal{P}_0) \cup U_j$ for a certain $j \in \{1, 2, 3\}$.

So suppose that $v \notin V_1'$. Let u be the only node in $V(\mathcal{T}) \cap V_1'$ such $V(P_{vu}) \cap V_1' = \{u\}$, where P_{vu} is the unique path in $G(\mathcal{T})$ from v to u. Let $\bar{P} = \mathcal{T} \cap (E(V_1') \cup [V_1', \{n_1, n_2\}]$. $\bar{P}$ is a path from u to one of the two nodes n_1 and n_2. Suppose, w.l.o.g., that $\bar{P}$ is from u to n_2. Since $u \in V_1$, there must exist a path $\mathcal{P}_u \in \mathcal{P}_{n_1 n_2}$ such that $u \in V(\mathcal{P}_u)$. Let $\mathcal{P}_u^1$ (resp., $\mathcal{P}_u^2$) be the path in $G(\mathcal{P}_u)$ from n_2 to u (resp., from u to n_1). We have $X(\mathcal{P}_u^1) = X(\bar{P})$ and $P^* = \bar{P} \cup \mathcal{P}_u^2 \in \mathcal{P}_{n_1 n_2}$. Thus, $T \in \mathcal{T}(X)$ (i.e., T is an A-tree such that $X(T) = 2$), where $T = (\mathcal{T} \setminus \bar{P}) \cup \mathcal{P}_u^1$. The equation $x(\mathcal{T}) = 2$ can be obtained from $x(T) = 2$, $x(\mathcal{P}_u) = 1$ and $x(P^*) = 1$. The result thus follows from Claim 7. □

Let S_1 (resp., S_2) be the set of nodes $v \in V \setminus (A \cup V(\mathcal{P}_0))$ such that there is some path $\mathcal{P}$ in $G \setminus A$ from v to some node of U_1 (resp., U_2) such that $V(\mathcal{P}) \cap V(\mathcal{P}_0) = \emptyset$. (We consider that $U_i \subset S_i$, $i = 1, 2$.) Let $S_3 = V \setminus (A \cup S_1 \cup S_2 \cup V(\mathcal{P}_0))$. Note that $V_3' \subset S_1$ and $V_2' \subset S_2$. Let $E_i = E(S_i \cup A \cup U_i \cup V(\mathcal{P}_0))$, $i = 1, 2, 3$. Thus, $E_i \cap E_j \subseteq E(V(\mathcal{P}_0))$ for $i, j \in \{1, 2, 3\}$, $i \neq j$. If $L \in \mathcal{P}^*(X) \cup \mathcal{T}^*(X)$, then by Claims 7 and 8, $L \subset E_k$ for a certain $k \in \{1, 2, 3\}$.

Claim 9. *$E(V(\mathcal{P}_0)) = \mathcal{P}_0$.*

Proof. Suppose that there exists $e = (uv) \in E(V(\mathcal{P}_0)) \setminus \mathcal{P}_0$. Let P be the unique path in $G(\mathcal{P}_0)$ from u to v. Since $(\mathcal{P}_0 \setminus P) \cup \{e\}$ is a path from n_1 to n_2, we have $X(e) \geq X(P)$. Since X is an extreme point of $\mathrm{LP}(G, A)$, there must exist $L \in \mathcal{P}^*(X) \cup \mathcal{T}^*(X)$ such that $e \in L$. Let $L_e = (L \setminus \{e\}) \cup P$. Since $X(L) = X(e) + X(L \setminus \{e\}) \leq X(L_e) \leq X(P) + X(L \setminus \{e\})$, we have $X(e) \leq X(P)$. Thus, $X(e) = X(P)$ and $L_e \in \mathcal{P}(X) \cup \mathcal{T}(X)$. Let $\bar{G}$ be the graph obtained from G by deleting e and $\bar{X}$ be the restriction of X to $E \setminus \{e\}$. Since $\bar{X} \in \mathrm{Q}(\bar{G}, A)$ and $\bar{X}(f) < 1$ for all $f \in E \setminus \{e\}$, then $\bar{X}$ is not an extreme point of $\mathrm{Q}(G, A)$. Otherwise, $\bar{X}$ also would be an extreme point of $\mathrm{LP}(\bar{G}, A)$, which is impossible since (G, A) is minimal. Hence, there must exist an extreme point $\bar{Y}$ of $\mathrm{Q}(\bar{G}, A)$ such that every tight constraint for $\bar{X}$ is also tight for $\bar{Y}$. Let X' be the solution defined by

$$X'(f) = \begin{cases} \bar{Y}(f) & \text{if } f \in E \setminus \{e\}, \\ \bar{Y}(P) & \text{if } f = e. \end{cases}$$

X' is also a solution of $S(X)$, which is impossible since $X' \neq X$. □

Let e_i be an edge of $E_i \setminus \mathcal{P}_0$, $i = 1, 2$, and e_0 be the only edge in $\delta(n_2) \cap \mathcal{P}_0$. Let X_i, $i = 1, 2$, be the solutions defined by

$$X_i(e) = \begin{cases} X(e) & \text{if } e \in E \setminus \{e_i\}, \\ 1 & \text{if } e = e_i. \end{cases}$$

Since (G, A) is a minimal pair and X_1 is not integral, it is not an extreme point of $\mathrm{LP}(G, A)$. Hence, there exist $t \geq 2$ integral extreme points $X_1^1, \ldots, X_1^t$ of $\mathrm{Q}(G, A)$ and t scalars $0 < \alpha_k < 1$, $k = 1, \ldots, t$, such that

$$X_1 = \sum_{k=1}^{t} \alpha_k X_1^k \qquad \text{and} \qquad \sum_{k=1}^{t} \alpha_k = 1.$$

Every tight constraint for X_1 is also tight for X_1^k, $k = 1, \ldots, t$. In particular, we have $X_1^k(\mathcal{P}_0) = 1$, $k = 1, \ldots, t$. Since $0 < X_1(e_0) < 1$, there must exist a certain $j \in \{1, \ldots, t\}$ such that $X_1^j(e_0) > 0$. Since X_1^j is integral, then $X_1^j(e_0) = 1$. Thus, $X_1^j(e) = 0$ for all $e \in \mathcal{P}_0 \setminus \{e_0\}$. W.l.o.g., suppose that $j = 1$. By the same arguments, there must exist an integral extreme point $X_2^1 \in \mathrm{Q}(G, A)$ such that every tight constraint by X_2 is also tight by X_2^1, $X_2^1(e_0) = 1$ and $X_2^1(e) = 0$ for all $e \in E(\mathcal{P}_0) \setminus \{e_0\}$. Let Y be the solution defined by

$$Y(e) = \begin{cases} X_1^1(e) & \text{if } e \in E_2 \cup E_3, \\ X_2^1(e) & \text{if } e \in E_1. \end{cases}$$

We can easily see that Y is also a solution of $S(X)$. Since $Y \neq X$, this contradicts the fact that X is the unique solution of $S(X)$. This ends the proof of our theorem.

Acknowledgments. I thank Hervé Kerivin for a careful reading of the paper and several valuable suggestions. I thank also the referees for their constructive comments. One of the referees was extraordinarily diligent in reading the original manuscript and provided many valuable suggestions that were incorporated in the final version.

REFERENCES

[1] G. Călinescu, H. Karloff, and Y. Rabani, *An improved approximation algorithm for MULTIWAY CUT*, in Proceedings of Symposium on Theory of Computing, ACM, 1998.

[2] S. Chopra and M. R. Rao, *On the multiway cut polyhedron*, Networks, 21 (1991), pp. 51–89.

[3] W. H. Cunningham, *The optimal multiterminal cut problem*, in Reliability of Computer and Communications Networks, C. Monma and F. Hwang, eds., AMS, Providence, 1991, pp. 105–120.

[4] W. H. Cunningham and L. Tang, *Optimal* 3*-terminal cuts and linear programming*, Lecture Notes in Comput. Sci. 1610, Springer, New York, 1999, pp. 114–125.

[5] E. Dahlhaus, D. Johnson, C. Papadimitriou, P. Seymour, and M. Yannakakis, *The complexity of multiterminal cuts*, SIAM J. Comput., 23 (1994), pp. 864–894.

[6] M. Grötschel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica, 1 (1981), pp. 70–89.

[7] H. S. Stone, *Multiprocessor scheduling with the aid of network flow algorithms*, IEEE Trans. Software Engrg., 3 (1977), pp. 85–93.

SIAM J. Discrete Math.
Vol. 19, No. 3, pp. 588–609

A BROOKS-TYPE THEOREM FOR THE GENERALIZED LIST T-COLORING*

JIŘÍ FIALA†, DANIEL KRÁĽ†, AND RISTE ŠKREKOVSKI‡

Abstract. We study the notion of a generalized list T-coloring which is a common generalization of the channel assignment problem and the T-coloring. An instance of the generalized list T-coloring is described by a triple (G, Λ, t), where G is a graph, Λ is a mapping which assigns the vertices of G lists of numbers (colors), and t is a mapping which assigns each edge of G a set of forbidden differences. We require that $0 \in t(e)$ for each edge e of G. The goal is to find a labeling c of the vertices of G with $c(v) \in \Lambda(v)$ for each vertex v, and $|c(u) - c(v)| \notin t(uv)$ for each edge uv of G. An instance is balanced if the size of the list $\Lambda(v)$ for each vertex v is equal to the sum of the sizes of $t(e)$ for edges e incident with v.

We state and prove a Brooks-type theorem for the generalized list T-coloring problem. This generalizes and unifies the previously known Brooks-type theorems for the channel assignment problem and for the T-coloring. The theorem characterizes balanced instances of the generalized list T-coloring with a good labeling. As a consequence, if G is a connected graph different from a Gallai tree, then all balanced instances on G have good labelings.

Key words. graph coloring, channel assignment problem, T-coloring, Brooks' theorem

AMS subject classification. 05C15

DOI. 10.1137/S0895480103437870

1. Introduction.

In this paper, we study a common generalization of the graph coloring, the list coloring, the T-coloring, and the (list) channel assignment problem. We call this coloring problem the *generalized list T-coloring*. Our approach unifies several previously known Brooks-type results, in particular [4, 10, 14, 22]. The addressed coloring problem was suggested by Hale [8] under the name of "frequency constrained channel assignment problem" as a model for assigning frequencies to radio transmitters. The generalized list T-coloring is a more flexible model for frequency assignment problems compared to the channel assignment problem because it has an additional capability: you may prevent every pair of transmitters from assigning frequencies which have certain special differences. E.g., the choice $T = \{0, 7, 14, 15\}$ gives a model for interferences for the UHF standard television transmitters [13].

An instance of the generalized list T-coloring is described by a triple (G, Λ, t) where G is a graph, $\Lambda : V(G) \to 2^{\mathbb{N}}$, and $t : E(G) \to 2^{\mathbb{N}}$, where $\mathbb{N}$ denotes the set of all nonnegative integers. In the rest, we write $V(G)$ and $E(G)$ for the vertex set and the edge set of a graph G. The elements of the sets $\Lambda(v)$ are called *colors* and the elements of the set $t(e)$ are called *forbidden differences* for the edge e. The function

*Received by the editors November 24, 2003; accepted for publication (in revised form) January 14, 2005; published electronically November 15, 2005.
http://www.siam.org/journals/sidma/19-3/43787.html

†Department of Applied Mathematics and Institute for Theoretical Computer Science (ITI), Faculty of Mathematics and Physics, Charles University, Malostranské náměstí 25, 118 00, Prague, Czech Republic (fiala@kam.mff.cuni.cz, kral@kam.mff.cuni.cz). The Institute for Theoretical Computer Science is supported by the Ministry of Education of Czech Republic as projects LN00A056 and 1M0021620808. The second author was a postdoctoral fellow at Technical University Berlin, Germany, from October 2004 to July 2005.

‡Department of Mathematics, University of Ljubljana, Jadranska 19, 1111 Ljubljana, Slovenia (skreko@kam.mff.cuni.cz). This research was partially done while this author was a postdoctoral fellow at Pacific Institute for the Mathematical Sciences (PIMS) at Simon Fraser University, Burnaby, BC, Canada.

t must satisfy $0 \in t(e)$ for each edge $e \in E(G)$ (this condition is more essential than it might seem at the first sight; see our concluding remarks in section 6, in particular an example given in Proposition 6.2). An instance of the generalized list T-coloring is also called a generalized list T-coloring problem. The goal of the problem is to find a mapping $c : V(G) \to \mathbb{N}$ with $c(v) \in \Lambda(v)$ for each $v \in V(G)$ and $|c(x) - c(y)| \notin t(xy)$ for each edge $xy \in E(G)$. Such a mapping is called a *good labeling.*

The *t-degree* $\deg_t(v)$ of a vertex v is equal to the sum $\sum_{vw \in E(G)} |t(vw)|$. An instance (G, Λ, t) is called *balanced* if $|\Lambda(v)| = \deg_t(v)$ holds for each vertex $v \in V(G)$. The problem is called *overbalanced* if $|\Lambda(v)| \geq \deg_t(v)$ holds for each $v \in V(G)$ and the inequality is strict for least at one vertex. The main result of this paper can be summarized as follows: every overbalanced instance of the generalized list T-coloring problem allows a good labeling (Theorem 3.1), and a complete description of balanced instances with no good labelings (Theorem 5.1) is provided. In particular, we show that if G is a connected graph distinct from a Gallai tree, then all balanced instances (G, Λ, t) allow a good labeling. Recall that a Gallai tree is a connected graph whose every block is an odd cycle or a complete graph.

We now explain how our results translate to the other coloring concepts mentioned above:

Graph coloring. An instance of the generalized list T-coloring problem corresponds to an instance of the usual graph k-coloring if Λ is the function constantly equal to $\{1, \ldots, k\}$ and t is the function constantly equal to $\{0\}$ at every edge. Theorem 3.1 translates to the well-known inequality $\chi(G) \leq \Delta(G) + 1$ and Theorem 5.1 to Brooks' theorem [4]. An elegant short proof of Brooks' theorem was given by Lovász [14]. An extension of Brooks' theorem to hypergraphs can be found in [9].

List coloring. The list coloring is a variant of the graph coloring where each vertex has to be assigned a color from its list [12, 20]. In our setting, Λ is just the function assigning lists of colors to the vertices and t is again a constant mapping equal to $\{0\}$ at every edge. Theorem 3.1 translates to the claim that each graph G is $(\Delta(G) + 1)$-choosable and Theorem 5.1 coincides with Brooks-type theorems for choosability and the list coloring from [2, 3, 6, 21]. Another theorem for the list coloring in the spirit of Brooks' theorem can be found in [11].

T-coloring and list T-coloring. In the T-coloring, the goal is to assign numbers (colors) to the vertices of a graph in such a way that the difference between the numbers assigned to two adjacent vertices does not belong to a certain fixed set of integers T (the set of forbidden differences); see [1, 13, 18, 23]. It is required that $0 \in T$. This condition assures that each T-coloring is also a coloring in the usual sense. In the list T-coloring, each vertex is also equipped with a list of available numbers (colors) and the assigned color must belong to the prescribed list. The generalized list T-coloring restricts to the T-coloring and the list T-coloring when the function t is a constant function equal to the set T. Our Theorem 5.1 for such a function t is just the Brooks-type theorem for the list T-coloring proved by Waller [22]: a 2-connected graph G is not $(|T| \cdot \Delta(G))$-T-choosable if and only if T is an arithmetic set and G is either a complete graph or an odd cycle. A set A of integers is called an *arithmetic set with a difference* d if $A = \{0, d, 2d, \ldots, d(k-1)\}$ for some integer k. Note that a set $\{0\}$ is arithmetic for all possible differences.

(List) channel assignment problem. Instances of the channel assignment problem are

graphs with edges labeled by positive integers. The numbers assigned to adjacent vertices must differ by at least the weight of the edge between the vertices [15]. The notion of the channel assignment problem also includes a so-called $L(p,q)$-labeling problem in which numbers assigned to adjacent vertices must differ by at least p and numbers assigned to vertices at distance two by at least q; see [5, 7, 19]. Theorem 3.1 translates to a counterpart of the inequality $\chi \le \Delta+1$ for the channel assignment problem proved by McDiarmid [16] and Theorem 5.1 extends the Brooks-type theorem for the list channel assignment problem from [10].

2. Preliminaries.

We write $A \uplus B$ for the union of disjoint sets A and B; this notation is used only to emphasize that the sets A and B are disjoint. Arithmetic sets are often considered in the paper, so we define $\mathrm{Ar}_d(k) = \{0, d, 2d, \ldots, d(k-1)\}$. For a set A of integers and an integer k_0, $A + k_0$ denotes the set $\{k + k_0 \mid k \in A\}$. If convenient, we use $k_0 + A$ instead of $A + k_0$. Similarly, $A - k_0$ denotes the set $\{k - k_0 \mid k \in A\}$ and $k_0 - A$ denotes the set $\{k_0 - k \mid k \in A\}$.

Let (G, Λ, t) be a generalized list T-coloring problem, v a vertex of a graph G, and α an element of $\Lambda(v)$. We say that the problem $(G', \Lambda', t') = (G, \Lambda, t)[v \to \alpha]$ is *obtained from the problem* (G, Λ, t) *by assigning the color* α *to the vertex* v. Formally, (G', Λ', t') is the following problem:

- $G' = G \setminus v$ is the subgraph of G induced by the vertex set $V(G) \setminus \{v\}$; i.e., $V(G') = V(G) \setminus \{v\}$ and $E(G') = \{ww' \mid ww' \in E(G) \;\&\; w, w' \in V(G')\}$.
- For each vertex w of G', the list $\Lambda'(w)$ is a subset of $\Lambda(w)$ consisting of the colors which do not conflict with the color assigning to the vertex v. Formally, $\Lambda'(w) = \{k \mid k \in \Lambda(w) \;\&\; |k - \alpha| \notin t(vw)\}$.
- The function t' is the restriction of the function t to $E(G')$; i.e., $t'(e) = t(e)$ for all $e \in E(G')$.

Clearly, the problem $(G', \Lambda', t') = (G, \Lambda, t)[v \to \alpha]$ has a good labeling if and only if the original problem (G, Λ, t) has a good labeling c with $c(v) = \alpha$. For a generalized list T-coloring problem (G, Λ, t), $\Lambda_{\min}$ and $\Lambda_{\max}$ denote the minimal and the maximal colors contained in the union $\bigcup_{v \in V(G)} \Lambda(v)$ of all lists.

The following lemma illustrates the just introduced notation.

LEMMA 2.1. *If* (G, Λ, t) *is a balanced generalized list* T*-coloring problem,* α *is* $\Lambda_{\min}$ *or* $\Lambda_{\max}$*, and* v *is an arbitrary vertex of* G *with* $\alpha \in \Lambda(v)$*, then the problem* $(G, \Lambda, t)[v \to \alpha]$ *is balanced or overbalanced. In particular, if there is a neighbor* v' *of* v *with that* $\alpha \notin \Lambda(v')$*, then* $(G, \Lambda, t)[v \to \alpha]$ *is overbalanced.*

Proof. By symmetry, it is enough to prove the lemma for $\alpha = \Lambda_{\min}$. The assignment of the color $\Lambda_{\min}$ to the vertex v reduces the size of the list $\Lambda(v')$ of each neighbor v' of the vertex v by at most $|t(vv')|$. Namely, only the elements of the set $t(vv') + \Lambda_{\min}$ can be removed. Observe that the t-degree of v' in $(G, \Lambda, t)[v \to \alpha]$ is $\deg_t(v') - |t(vv')|$. Thus, if (G, Λ, t) is balanced and $t(vv') + \Lambda_{\min} \subseteq \Lambda(v')$ for each neighbor v' of v, then the new problem is balanced, too. If the latter condition is not satisfied for some neighbor v' of v, then the size of the list of v' is decreased by at most $|t(vv')| - 1$, and thus the new problem is overbalanced. In particular, this happens if $\Lambda_{\min} \notin \Lambda(v')$. □

3. The counterpart of the inequality $\chi \le \Delta + 1$.

In this section, we prove the counterpart of the well-known graph inequality $\chi \le \Delta + 1$.

THEOREM 3.1. *An overbalanced generalized list* T*-coloring problem* (G, Λ, t) *has a good labeling whenever* G *is a connected graph.*

Proof. The proof is by induction on the number of vertices of G. If $|V(G)| = 1$, then $\Lambda(v) \neq \emptyset$ for the single vertex v of G, and hence (G, Λ, t) has a good labeling. Assume in the rest that $|V(G)| \geq 2$. Let $V_{\min}$ be the set of vertices v of G such that $\Lambda_{\min} \in \Lambda(v)$, and let v_0 be a vertex of G with $\deg_t(v_0) < |\Lambda(v_0)|$. In the proof, we distinguish three cases with respect to the vertex v_0 and the set $V_{\min}$.

If $V_{\min}$ contains a vertex v which is not a cut-vertex of G and $v \neq v_0$, then assign the color $\Lambda_{\min}$ to the vertex v and obtain an overbalanced problem $(G', \Lambda', t') = (G, \Lambda, t)[v{\rightarrow}\Lambda_{\min}]$. Since G' is connected, the problem (G', Λ', t') has a good labeling by the induction hypothesis. Hence, the problem (G, Λ, t) has a good labeling, too.

If the first case does not hold and $|V_{\min}| \geq 2$, it can be easily seen that $V_{\min}$ contains a cut vertex v, $v \neq v_0$, with the following property: if K is the component of $G \setminus v$ which contains the vertex v_0, then each component of $G \setminus v$ distinct from K contains no vertices of $V_{\min}$. Consider now the problem (G', Λ', t') obtained by assigning the color $\Lambda_{\min}$ to the vertex v. The problem (G', Λ', t') restricted to the component K is overbalanced because K contains the vertex v_0. The corresponding problems obtained by restricting to the other components are also overbalanced: each of the other components contains a neighbor of v whose list $\Lambda(v)$ does not contain the color $\Lambda_{\min}$. Each of these restricted problems is overbalanced by Lemma 2.1 and its underlying graph is connected. Hence, they all have good labelings, and thus the generalized list T-coloring problem (G, Λ, t) also has a good labeling.

The remaining case is $V_{\min} = \{v_0\}$. Consider now the problem (G', Λ', t') obtained by assigning the color $\Lambda_{\min}$ to v_0 and its restrictions to all the components of $G \setminus v_0$. Each of these restrictions is overbalanced by Lemma 2.1 because v_0 is the only vertex whose list contains the color $\Lambda_{\min}$. By the induction hypothesis, all of them have good labelings, and thus the problem (G, Λ, t) has a good labeling, too. □

4. The case of 2-connected graphs.

In this section, we characterize balanced generalized list T-coloring problems (G, Λ, t) with 2-connected graphs G which have no good labelings. These results are then used in section 5 where a characterization of all balanced generalized list T-coloring problems with no good labeling is presented.

LEMMA 4.1. *Let (G, Λ, t) be a balanced generalized list T-coloring problem such that G is 2-connected. If there is a vertex v such that $\Lambda_{\min} \notin \Lambda(v)$ or $\Lambda_{\max} \notin \Lambda(v)$, then the problem (G, Λ, t) has a good labeling.*

Proof. By symmetry, it is enough to prove the lemma for the case that $\Lambda_{\min}$ is not contained in all lists. In such case, since G is connected, there must be adjacent vertices v and w such that $\Lambda_{\min} \notin \Lambda(v)$ and $\Lambda_{\min} \in \Lambda(w)$. The problem $(G, \Lambda, t)[w{\rightarrow}\Lambda_{\min}]$ is overbalanced by Lemma 2.1. And since $G \setminus w$ is a connected graph, it follows from Theorem 3.1 that $(G, \Lambda, t)[w{\rightarrow}\Lambda_{\min}]$ has a good labeling. Thus, the original problem (G, Λ, t) has a good labeling, too. □

The following well-known lemma can be found in [17, Lemma 1.15].

LEMMA 4.2. *Every 2-connected graph G, which is neither a cycle nor a complete graph, contains three vertices x, y, and z such that x and y are neighbors of z, the vertices x and y are nonadjacent, and the graph $G \setminus \{x, y\}$ is connected.*

Lemma 4.2 allows us to focus on the problems where G is either an odd cycle or a complete graph. The next lemma deals with balanced generalized list T-coloring problems whose underlying graphs are 2-connected but they are neither odd cycles nor complete graphs. The cases of odd cycles and complete graphs are considered in separate subsections later.

LEMMA 4.3. *If a balanced generalized list T-coloring problem (G, Λ, t) does not have a good labeling and G is 2-connected, then G is either an odd cycle or a complete* *graph.*

Proof. Suppose G is neither an odd cycle nor a complete graph and (G, Λ, t) does not have a good labeling. By Lemma 4.1, the color $\Lambda_{\min}$ is contained in the list $\Lambda(v)$ for every vertex $v \in V(G)$. Let us first consider the case that G is an even cycle. Let $v_1, \ldots, v_n$ be the vertices of the cycle G enumerated in a cyclic order. Let k_i be a color of $\Lambda(v_i) \setminus ((t(v_{i-1}v_i) + \Lambda_{\min}) \cup (t(v_iv_{i+1}) + \Lambda_{\min}))$ for each $1 \leq i \leq n$. Since the problem is balanced (recall that $0 \in t(v_{i-1}v_i) \cap t(v_iv_{i+1})$), such a number k_i always exists. Then we can define a good labeling c as follows:

$$c(v_i) = \begin{cases} \Lambda_{\min} & \text{if } i \text{ is odd,} \\ k_i & \text{otherwise.} \end{cases}$$

The remaining case is that the graph G is neither a complete graph nor a cycle. Let x, y, and z be vertices of G with the properties as described in the statement of Lemma 4.2. Recall that the color $\Lambda_{\min}$ is contained in the list $\Lambda(v)$ of every vertex $v \in V(G)$. Consider now the problem (G', Λ', t') obtained from (G, Λ, t) by assigning the color $\Lambda_{\min}$ to the vertices x and y. By Lemma 2.1, the problem $(G, \Lambda, t)[x{\rightarrow}\Lambda_{\min}]$ is balanced and the color $\Lambda_{\min}$ for $(G, \Lambda, t)[x{\rightarrow}\Lambda_{\min}]$ is not contained in the list of z. Note that z is a neighbor of y. Hence, the problem $(G', \Lambda', t') = ((G, \Lambda, t)[x{\rightarrow}\Lambda_{\min}])[y{\rightarrow}\Lambda_{\min}]$ is overbalanced by Lemma 2.1. However, the problem (G', Λ', t') has a good labeling by Theorem 3.1, and thus the original problem (G, Λ, t) has a good labeling, too. □

The following lemma is a corollary of the Brooks-type theorem for the T-coloring proved by Waller [22, Lemma 7]. We provide here its complete proof for the sake of completeness.

LEMMA 4.4. *If (K_2, Λ, t) is a balanced generalized list T-coloring problem with $K_2 = uv$, then (K_2, Λ, t) admits no good labeling if and only if $t(uv)$ is arithmetic and $\Lambda(u) = \Lambda(v) = \Lambda_{\min} + t(uv)$.*

Proof. By Lemma 4.1, it holds that $\Lambda_{\min} \in L(u)$ and $\Lambda_{\min} \in L(v)$. If $\Lambda(u) \neq \Lambda_{\min} + t(uv)$, then there is a good labeling which assigns $\Lambda_{\min}$ to v and a color of $\Lambda(u) \setminus (\Lambda_{\min} + t(uv))$ to u. Hence, $\Lambda(u) = \Lambda_{\min} + t(uv)$, and similarly, $\Lambda(v) = \Lambda_{\min} + t(uv)$.

If $t(uv)$ is not arithmetic, then K_2 has a good labeling from any pair of lists of size $|t(uv)|$; let $0 = i_1 < i_2 < \cdots < i_k$ be the elements of $t(uv) = \Lambda(u) - \Lambda_{\min} = \Lambda(v) - \Lambda_{\min}$ and let k_0 be the largest index such that the set $\{i_1, \ldots, i_{k_0}\}$ is arithmetic. Since $t(uv)$ is not arithmetic, we have $2 \leq k_0 < k$. Observe now that $i_{k_0+1} - i_2 \notin t(uv)$ by the choice of k_0. However, the labeling c that is defined as $c(u) = \Lambda_{\min} + i_2$ and $c(v) = \Lambda_{\min} + i_{k_0+1}$ is good. □

4.1. The case of odd cycles.

Throughout this subsection, we consider cycles C_n of odd length n. The vertices of a cycle C_n are denoted by $v_1, \ldots, v_n$. In the next lemma, we study a possible structure of sets $t(e)$ in balanced generalized list T-coloring problems (C_n, Λ, t) with no good labelings.

LEMMA 4.5. *Let (C_n, Λ, t) be a balanced generalized list T-coloring problem where C_n is an odd cycle. If the set $t(e)$ is not arithmetic for an edge e of C_n or $\Lambda_{\max} - \Lambda_{\min} \in t(e)$, then the problem (C_n, Λ, t) has a good labeling.*

Proof. Suppose that (C_n, Λ, t) does not have a good labeling. We may assume that the colors $\Lambda_{\min}$ and $\Lambda_{\max}$ are contained in all the lists $\Lambda(v)$, $v \in V(C_n)$ by Lemma 4.1. Assume that the edge e from the statement of the lemma is the edge

v_1v_2. We first define a sought good labeling c for vertices $v_3, \ldots, v_n$ as follows:

$$\begin{array}{lcll} c(v_i) & = & \Lambda_{\min} & \text{for } i = 3, 5, \ldots, n \text{ and} \\ c(v_i) & \in & \Lambda(v_i) \setminus ((\Lambda_{\min} + t(v_{i-1}v_i)) \cup (\Lambda_{\min} + t(v_{i+1}v_i))) & \text{for } i = 4, 6, \ldots, n-1. \end{array}$$

Note that the set $\Lambda(v_i) \setminus ((\Lambda_{\min} + t(v_{i-1}v_i)) \cup (\Lambda_{\min} + t(v_{i+1}v_i)))$ is nonempty for each $i = 4, 6, \ldots, n-1$ because the problem (C_n, Λ, t) is balanced and $0 \in t(v_{i-1}v_i) \cap t(v_{i+1}v_i)$.

Consider now the problem (G', Λ', t') obtained by assigning the color $c(v_i)$ to every vertex v_i for $i = 3, \ldots, n$. Note that G' is isomorphic to K_2 (it is just the edge v_1v_2) and the problem (G', Λ', t') is balanced or overbalanced (follow the proof of Lemma 4.1). If $t(e)$ is not arithmetic, then the problem (G', Λ', t') has a good labeling by Lemma 4.4. Otherwise, $\Lambda_{\max} - \Lambda_{\min} \in t(e) = t'(e)$ by the assumption of the lemma. Since the vertices v_3 and v_n are colored with $\Lambda_{\min}$, the colors contained in the list $\Lambda'(v_1)$ and $\Lambda'(v_2)$ are integers between $\Lambda_{\min} + 1$ and $\Lambda_{\max}$. Hence, the problem (G', Λ', t') has a good labeling by Lemma 4.4 in this case, too. Thus, the original problem (G, Λ, t) has a good labeling in both the cases. □

The following lemma relates contents of lists $\Lambda(v)$ to sets $t(e)$ for balanced generalized list T-coloring problems (C_n, Λ, t) with no good labelings.

LEMMA 4.6. *If (C_n, Λ, t) is a balanced generalized list T-coloring problem with no good labeling, then the following equalities hold for each i, $1 \le i \le n$ (indices are taken modulo n):*

$$\Lambda(v_i) = (\Lambda_{\min} + t(v_{i-1}v_i)) \uplus (\Lambda_{\max} - t(v_iv_{i+1})) \tag{4.1}$$

$$= (\Lambda_{\min} + t(v_iv_{i+1})) \uplus (\Lambda_{\max} - t(v_{i-1}v_i)). \tag{4.2}$$

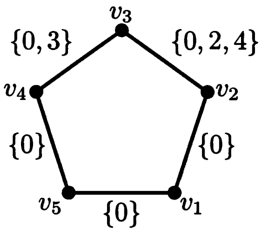
Proof. By Lemma 4.1, each list $\Lambda(v_i)$ contains both the colors $\Lambda_{\min}$ and $\Lambda_{\max}$. In addition, by Lemma 4.5, there is no edge $e = v_iv_{i+1}$ with $\Lambda_{\max} - \Lambda_{\min} \in t(e)$. Suppose that there is a vertex v_i whose list $L(v_i)$ does not satisfy the equality of (4.1), i.e., either the sets $\Lambda_{\min} + t(v_{i-1}v_i)$ and $\Lambda_{\max} - t(v_iv_{i+1})$ are not disjoint or $\Lambda(v_i) \neq (\Lambda_{\min} + t(v_{i-1}v_i)) \uplus (\Lambda_{\max} - t(v_iv_{i+1}))$. Since the problem is balanced, the size of the list $L(v_i)$ is $|t(v_{i-1}v_i)| + |t(v_iv_{i+1})|$ and there is a color k such that $k \in \Lambda(v_i) \setminus ((\Lambda_{\min} + t(v_{i-1}v_i)) \cup (\Lambda_{\max} - t(v_iv_{i+1})))$ in both the cases. Consider now the following labeling c (indices are taken modulo n):

$$c(v_j) = \begin{cases} k & \text{for } j = i, \\ \Lambda_{\min} & \text{for } j = i+2, i+4, \ldots, i-1, \\ \Lambda_{\max} & \text{for } j = i+1, i+3, \ldots, i-2. \end{cases}$$

The labeling c is good by the choice of the color k and the fact that $\Lambda_{\max} - \Lambda_{\min} \notin t(e)$ for all edges e of the cycle. The equality (4.2) can be proven analogously. □

Before stating Theorem 4.7, we define three special types of vertices for the problems whose underlying graphs are cycles. Let (C_n, Λ, t) be a balanced generalized list T-coloring problem. We say that the vertex v_i is of the *first*, *second*, or *third type* if it satisfies the following condition 1, 2, or 3, respectively:

1. $t(v_{i-1}v_i) = t(v_iv_{i+1})$ is arithmetic and $\Lambda(v_i) = (\Lambda_{\min} + t(v_{i-1}v_i)) \uplus (\Lambda_{\max} - t(v_{i-1}v_i))$.
2. The sets $t(v_{i-1}v_i)$ and $t(v_iv_{i+1})$ are arithmetic with the same difference d but $t(v_{i-1}v_i) \neq t(v_iv_{i+1})$. The list $\Lambda(v_i)$ is $\Lambda_{\min} + \mathrm{Ar}_d(k)$, where $k = |t(v_{i-1}v_i)| + |t(v_iv_{i+1})|$. In particular, $\Lambda_{\max} - \Lambda_{\min} = d(k-1)$.



$$\Lambda(v_1) = \{2,8\}$$
$$\Lambda(v_2) = \{2,4,6,8\}$$
$$\Lambda(v_3) = \{2,4,5,6,8\}$$
$$\Lambda(v_4) = \{2,5,8\}$$
$$\Lambda(v_5) = \{2,8\}$$

FIG. 4.1. *An example of a balanced generalized list T-coloring problem with an underlying graph being C_5. The sets of forbidden differences are at the centers of the corresponding edges and the lists of colors for the vertices are on the right.*

3. Both $t(v_{i-1}v_i)$ and $t(v_iv_{i+1})$ are arithmetic sets with at least two elements, and their differences d and d' are distinct. Then $t(v_{i-1}v_i) = \mathrm{Ar}_d(k)$ and $t(v_iv_{i+1}) = \mathrm{Ar}_{d'}(k')$, where $kd = k'd' = \mathrm{lcm}(d, d')$. In addition, $\Lambda_{\max} - \Lambda_{\min} = \mathrm{lcm}(d, d')$ and

$$\Lambda(v_i) = (\Lambda_{\min}+\mathrm{Ar}_d(k))\uplus(\Lambda_{\max}-\mathrm{Ar}_{d'}(k')) = (\Lambda_{\min}+\mathrm{Ar}_{d'}(k'))\uplus(\Lambda_{\max}-\mathrm{Ar}_d(k)).$$

Note that both the unions in the above expression are disjoint because of the equality $kd = k'd' = \mathrm{lcm}(d, d') = \Lambda_{\max} - \Lambda_{\min}$.

As an example, consider the generalized list T-coloring problem depicted in Figure 4.1. The vertices v_1 and v_5 are of the first type, the vertices v_2 and v_4 are of the second type, and the vertex v_3 is of the third type. Note that the problem depicted in Figure 4.1 has no good labeling.

We finally characterize balanced generalized list T-coloring problems (C_n, Λ, t) with no good labelings.

THEOREM 4.7. *A balanced generalized list T-coloring problem (C_n, Λ, t), where C_n is an odd cycle, does not have a good labeling if and only if*

- *the colors $\Lambda_{\min}$ and $\Lambda_{\max}$ are contained in all the lists $\Lambda(v)$, $v \in V(C_n)$,*
- *each vertex is one of the three types described above; in particular, all the sets $t(e)$, $e \in E(C_n)$, are arithmetic, and*
- *there is at least one vertex of the first or of the second type.*

Proof. We first prove that if a balanced problem (C_n, Λ, t) does not have a good labeling, then it is of the form described in the statement. The colors $\Lambda_{\min}$ and $\Lambda_{\max}$ are contained in all the lists by Lemma 4.1, and all the sets $t(e)$, $e \in E(C_n)$ are arithmetic by Lemma 4.5. Fix an arbitrary vertex v_i of C_n. We show that the vertex v_i is one of the three types introduced before this theorem.

If $t(v_{i-1}v_i) = t(v_iv_{i+1})$, then $\Lambda(v_i) = (\Lambda_{\min} + t(v_{i-1}v_i)) \uplus (\Lambda_{\max} - t(v_{i-1}v_i))$ by Lemma 4.6. Hence, the vertex v_i is of the first type.

We may now assume that $t(v_{i-1}v_i) \neq t(v_iv_{i+1})$. Let $t(v_{i-1}v_i) = \mathrm{Ar}_d(k)$ and $t(v_iv_{i+1}) = \mathrm{Ar}_{d'}(k')$. If $k = 1$ or $k' = 1$, i.e., the set assigned to the corresponding edge incident with v_i is $\{0\}$, then we may assume that the differences d and d' are equal. However, if $d = d'$, then, by Lemma 4.6 we have the following:

$$\Lambda(v_i) = (\Lambda_{\min} + \mathrm{Ar}_d(k)) \uplus (\Lambda_{\max} - \mathrm{Ar}_d(k')) = (\Lambda_{\min} + \mathrm{Ar}_d(k')) \uplus (\Lambda_{\max} - \mathrm{Ar}_d(k)).$$

But this is possible only if $\Lambda_{\max} - \Lambda_{\min} = d(k + k' - 1)$. Hence, the vertex v_i is of the second type.

The final case is that $d \neq d'$, say $d < d'$, and both k and k' are at least 2. By Lemma 4.6, we have:

$$\Lambda(v_i) = (\Lambda_{\min} + \mathrm{Ar}_d(k)) \uplus (\Lambda_{\max} - \mathrm{Ar}_{d'}(k')) = (\Lambda_{\min} + \mathrm{Ar}_{d'}(k')) \uplus (\Lambda_{\max} - \mathrm{Ar}_d(k)).$$

But this is possible only if $\Lambda_{\max} - \Lambda_{\min} = \mathrm{lcm}(d, d') = kd = k'd'$. Indeed, the set $\Lambda(v_i)$ contains the element $\Lambda_{\min} + d$ by the middle part of the equality. Since $d < d'$, then $\Lambda_{\min} + d$ must be equal to $\Lambda_{\max} - (k-1)d$ by the right part of the equality. We now have $\Lambda_{\max} - \Lambda_{\min} = kd$, as desired. Since the unions in the above equality are disjoint, we have also $\Lambda_{\max} - \Lambda_{\min} = k'd'$ and $\mathrm{lcm}(d, d') = kd = k'd'$. Hence, we have inferred that the vertex v_i is of the third type.

In order to complete the proof of the first implication of the theorem, it remains to exclude the case that all the vertices are of the third type. So, assume now that all the vertices are of the third type. Let d_i be the difference of the arithmetic set $t(v_iv_{i+1})$. Consider the labeling c defined as $c(v_i) = \Lambda_{\min} + d_i$ for each $i = 1, \ldots, n$. Since $\Lambda_{\min} + d_i \in \Lambda(v_i)$, the labeling cannot be a good labeling only if there is an index i such that $|(\Lambda_{\min} + d_{i+1}) - (\Lambda_{\min} + d_i)| = |d_{i+1} - d_i| \in t(v_iv_{i+1})$. Then $d_i|(d_{i+1} - d_i)$ and $d_i|d_{i+1}$. Hence, $\mathrm{lcm}(d_i, d_{i+1}) = d_{i+1}$ and $t(v_{i+1}v_{i+2}) = \mathrm{Ar}_{d_{i+1}}(1)$. But then, the vertex v_{i+1} is not of the third type.

We now prove the opposite implication of the theorem, namely, that a balanced generalized list T-coloring problem of the form described in the theorem does not have a good labeling. The proof proceeds by contradiction, which is eventually obtained when several claims have been established. Let c be a good labeling of such a problem (C_n, Λ, t) and let d_i be the difference of the arithmetic set $t(v_iv_{i+1})$. We construct another function $\mu : V(C_n) \to \mathbb{N} \cup \{\mathrm{Min}, \mathrm{Max}\}$ based on the labeling c:

$$\mu(v_i) = \begin{cases} \mathrm{Min} & \text{if } c(v_i) \in (\Lambda_{\min} + t(v_{i-1}v_i)) \cap (\Lambda_{\min} + t(v_iv_{i+1})), \\ \mathrm{Max} & \text{if } c(v_i) \in (\Lambda_{\max} - t(v_{i-1}v_i)) \cap (\Lambda_{\max} - t(v_iv_{i+1})), \\ d_{i-1} & \text{if } c(v_i) \in (\Lambda_{\min} + t(v_{i-1}v_i)) \setminus (\Lambda_{\min} + t(v_iv_{i+1})), \qquad (*) \\ d_i & \text{if } c(v_i) \in (\Lambda_{\min} + t(v_iv_{i+1})) \setminus (\Lambda_{\min} + t(v_{i-1}v_i)). \qquad (**) \end{cases}$$

Since all the vertices are of one of the three types, all the lists $\Lambda(v_i)$ satisfy the equalities (4.1) and (4.2) from Lemma 4.6. Hence, the function μ is well defined. Observe that if v_i is of the first type (in which $t(v_{i-1}v_i) = t(v_iv_{i+1})$), then $\Lambda_{\min} + t(v_{i-1}v_i) = \Lambda_{\min} + t(v_iv_{i+1})$. Hence, $\mu(v_i)$ for such a vertex v_i is either Min or Max. In particular, we have the following.

CLAIM 4.7.1. *If $\mu(v_i) \notin \{\mathrm{Min}, \mathrm{Max}\}$, then v_i is of the second type or the third type.*

We now prove the following two claims.

CLAIM 4.7.2. *If c is a good labeling, then no two adjacent vertices are simultaneously assigned by μ both the label* Min *or both the label* Max.

If two adjacent vertices v_i and v_{i+1} are both mapped to Min, then $c(v_i) \in (\Lambda_{\min} + t(v_iv_{i+1}))$ and $c(v_{i+1}) \in (\Lambda_{\min} + t(v_iv_{i+1}))$ by the definition of μ. This immediately yields that $|c(v_i) - c(v_{i+1})| \in t(v_iv_{i+1})$ (recall that the set $t(v_iv_{i+1})$ is arithmetic). A similar argument excludes the case that both v_i and v_{i+1} are mapped to Max.

CLAIM 4.7.3. *If the vertex v_i is assigned by μ the difference d_{i-1}, i.e., the condition in $(*)$ is satisfied, then $d_{i-1}|\Lambda_{\max} - \Lambda_{\min}$ and the following holds:*

$$\{\Lambda_{\min}, \Lambda_{\min} + d_{i-1}, \Lambda_{\min} + 2d_{i-1}, \ldots, \Lambda_{\max}\} \subseteq (c(v_i) - t(v_{i-1}v_i)) \cup (c(v_i) + t(v_{i-1}v_i)). \qquad (\triangle)$$

Since v_i is assigned by μ neither Min nor Max, the vertex v_i is of the second type or the third type by Claim 4.7.1. Hence, $d_{i-1}|\Lambda_{\max} - \Lambda_{\min}$. If v_i is of the third type, then $t(v_{i-1}v_i) = \mathrm{Ar}_{d_{i-1}}(k)$, where $k = (\Lambda_{\max} - \Lambda_{\min})/d_{i-1} - 1$. Since $\mu(v_i)$ is neither Min nor Max, the color $c(v_i)$ is neither $\Lambda_{\min}$ nor $\Lambda_{\max}$. We infer from $c(v_i) \neq \Lambda_{\min}, \Lambda_{\max}$ that $c(v_i) \in \{\Lambda_{\min} + d_{i-1}, \Lambda_{\min} + 2d_{i-1}, \ldots, \Lambda_{\max} - d_{i-1}\}$. Hence, the inclusion ($\triangle$) indeed holds.

Next, we consider the case in which v_i is of the second type. Let $k_{i-1} = |t(v_{i-1}v_i)|$ and $k_i = |t(v_i v_{i+1})|$. Note that $k_{i-1} > k_i$ by ($*$) and $\Lambda_{\max} - \Lambda_{\min} = (k_{i-1} + k_i - 1)d_{i-1}$ because v_i is of the second type. By the condition from ($*$), the color $c(v_i)$ is one of the numbers $\Lambda_{\min} + k_i d_i, \Lambda_{\min} + (k_i + 1)d_i, \ldots, \Lambda_{\min} + (k_{i-1} - 1)d_i$ and thus the inclusion ($\triangle$) holds in this case, too. This establishes the claim.

Similar to the proof of Claim 4.7.3, we can prove the following claim (the details are left to the reader).

CLAIM 4.7.4. *If the vertex v_i is assigned by μ the difference d_i, i.e., the condition in ($**$) is satisfied, then $d_i|\Lambda_{\max} - \Lambda_{\min}$ and the following holds:*

$$\{\Lambda_{\min}, \Lambda_{\min}+d_i, \Lambda_{\min}+2d_i, \ldots, \Lambda_{\max}\} \subseteq (c(v_i)-t(v_i v_{i+1})) \cup (c(v_i)+t(v_i v_{i+1})). \quad (\triangle\triangle)$$

Now, some edges of the cycle are oriented in the following way: if v_i is labeled by μ with d_{i-1} according to ($*$), then the edge $v_{i-1}v_i$ is oriented from v_i to v_{i-1}. If v_i is labeled by μ with d_i according to ($**$), then the edge $v_i v_{i+1}$ is oriented from v_i to v_{i+1}. Since c is a good labeling, each edge is oriented in at most one direction. Indeed, assume for the sake of contradiction that both the vertex v_i satisfies ($**$) and the vertex v_{i+1} satisfies ($*$). We can now infer from ($\triangle\triangle$) that $d_i|(c(v_i) - \Lambda_{\min})$; in particular, $c(v_i) \in \{\Lambda_{\min}, \Lambda_{\min}+d_i, \Lambda_{\min}+2d_i, \ldots, \Lambda_{\max}\}$ holds. Since the vertex v_{i+1} satisfies ($\triangle$), we conclude that $|c(v_{i+1}) - c(v_i)| \in t(v_i v_{i+1})$.

The proof of the second implication is completed by the following four claims.

CLAIM 4.7.5. *No edge can be oriented to a vertex which is assigned by μ either* Min *or* Max.

Assume the opposite and say, e.g., that the edge $v_i v_{i+1}$ is oriented from v_i to v_{i+1} and $\mu(v_{i+1}) = \text{Min}$. Then $c(v_i) \in (\Lambda_{\min} + t(v_i v_{i+1})) \setminus (\Lambda_{\min} + t(v_{i-1}v_i))$ and the vertex v_i is assigned by μ the difference d_i. In particular, $\Lambda_{\max} - \Lambda_{\min}$ is divisible by d_i and $(c(v_i) + t(v_i v_{i+1})) \cup (c(v_i) - t(v_i v_{i+1})) \supseteq \Lambda_{\min} + \mathrm{Ar}_{d_i}(k+1)$ by ($\triangle\triangle$), where $k = (\Lambda_{\max} - \Lambda_{\min})/d_i$. Since the vertex v_{i+1} is assigned by μ the label Min, the difference $c(v_{i+1}) - \Lambda_{\min}$ is divisible by d_i and thus $c(v_{i+1}) \in \Lambda_{\min} + \mathrm{Ar}_{d_i}(k+1)$. Then $|c(v_i) - c(v_{i+1})| \in t(v_i v_{i+1})$ and the labeling c is not good.

CLAIM 4.7.6. *All edges of the cycle are oriented.*

If all the vertices of the cycle are assigned by μ one of the labels Min or Max, then the vertices of the cycle should be assigned the labels Min and Max alternately. But this is impossible because the length of the cycle is odd. Hence, there is a vertex v_i assigned by μ neither Min nor Max. In particular, there is an edge leaving the vertex v_i and this edge must lead to a vertex which is again assigned by μ neither Min nor Max by Claim 4.7.5. There is also an edge leaving this vertex and it again leads to a vertex assigned by μ neither Min nor Max. In this way, we go around the whole cycle and show that all the edges are oriented.

CLAIM 4.7.7. *All the vertices of the cycle are of the second type or the third type.*

By Claim 4.7.6, all edges of the cycle are oriented. Since no edge can be oriented to a vertex which is assigned by μ either Min or Max by Claim 4.7.5, all the vertices are of the second or the third type by Claim 4.7.1.

Claim 4.7.8. *All the vertices of the cycle are of the third type.*

Assume that the vertex v_i is of the second type. By symmetry, it can be assumed that $t(v_{i-1}v_i) \subseteq t(v_iv_{i+1})$. Since v_i is of the second type, it holds that $t(v_{i-1}v_i) \subset t(v_iv_{i+1})$. Now let k_{i-1} and k_i be such integers that $t(v_{i-1}v_i) = \mathrm{Ar}_{d_{i-1}}(k_{i-1})$ and $t(v_iv_{i+1}) = \mathrm{Ar}_{d_i}(k_i)$. Note that $k_{i-1} < k_i$ and $k_{i-1} + k_i = (\Lambda_{\max} - \Lambda_{\min})/d_{i-1} + 1$. In particular, $k_{i-1} < (\Lambda_{\max} - \Lambda_{\min})/d_{i-1}$. Since all the edges are oriented, the edge $v_{i-1}v_i$ is oriented from v_{i-1} to v_i. If the vertex v_{i-1} were of the third type, then it would hold that $k_{i-1} = (\Lambda_{\max} - \Lambda_{\min})/d_{i-1}$ (by the definition of the third vertex type). However, this does not hold. Hence, v_{i-1} is of the second type and $t(v_{i-2}v_{i-1}) = t(v_iv_{i+1}) = \mathrm{Ar}_{d_i}(k_i)$. But then the edge $v_{i-1}v_i$ cannot be oriented from v_{i-1} to v_i because $t(v_{i-1}v_i) \subseteq t(v_{i-2}v_{i-1})$. This establishes the claim.

By Claim 4.7.8, all the vertices are of the third type, but then the balanced generalized list T-coloring problem is not as described in the statement of the theorem. This completes the proof of the second implication and so the proof of the whole theorem. □

4.2. The case of complete graphs. We first formulate a lemma that is an immediate corollary of Theorem 4.7 but that is useful in the analysis of the case of complete graphs.

Lemma 4.8. *Let (C_3, Λ, t) be a balanced generalized list T-coloring problem and let $V(C_3) = \{x, y, z\}$. If (C_3, Λ, t) does not have a good labeling, then the sets $t(xy)$, $t(xz)$, and $t(yz)$ are arithmetic. Moreover, if each of the sets $t(xy)$ and $t(xz)$ contains at least two elements and the differences of the arithmetic sets $t(xy)$ and $t(xz)$ are distinct, then $t(yz) = \{0\}$.*

Proof. Since the problem (C_3, Λ, t) does not have a good labeling, it is of the type described in the statement of Theorem 4.7. Therefore, the sets $t(xy)$, $t(xz)$, and $t(yz)$ are arithmetic. If each of the arithmetic sets $t(xy)$ and $t(xz)$ contains at least two elements and their differences are distinct, then the vertex x must be of the third type. Consequently, at least one of the vertices y and z is of the first or the second type by Theorem 4.7. Assume that this vertex is y. Then the difference of the arithmetic set $t(yz)$ and the difference of the arithmetic set $t(yx)$ are the same. Let d be this difference. Since x is of the third type, we have $t(yx) = \mathrm{Ar}_d((\Lambda_{\max} - \Lambda_{\min})/d)$ by the definition of the third type. By Lemma 4.6, the sets $\Lambda_{\min} + t(yx)$ and $\Lambda_{\max} - t(yz)$ are disjoint. But this is possible only if $t(yz) = \{0\}$ (recall that the difference of $t(yz)$ is d). □

Next, we show that a balanced generalized list T-coloring problem with no good labeling can be reduced to a smaller one with the same property.

Lemma 4.9. *Let (K_n, Λ, t) be a balanced generalized list T-coloring problem with no good labeling. If U be a subset of $V(K_n)$ of size $n' \geq 2$, then there is a balanced generalized list T-coloring problem $(K_{n'}, \Lambda', t')$ which does not have a good labeling, $V(K_{n'}) = U$, and $t'(uu') = t(uu')$ for $u, u' \in U$.*

Proof. The proof proceeds by induction on $n - n'$. If $n - n' = 0$, then the problems (K_n, Λ, t) and $(K_{n'}, \Lambda', t')$ are the same.

Assume $n - n' = 1$. Since the problem (K_n, Λ, t) does not have a good labeling, it follows that the color $\Lambda_{\min}$ is contained in each list $\Lambda(v)$ by Lemma 4.1. Consider the problem $(K_n, \Lambda, t)[v \rightarrow \Lambda_{\min}]$ where v is the only vertex of K_n outside the set U. In particular, the problem $(K_n, \Lambda, t)[v \rightarrow \Lambda_{\min}]$ is balanced and it does not have a good labeling.

If $n - n' \geq 2$, consider a set U' of the vertices of K_n such that $U \subset U' \subset V(K_n)$. By the induction hypothesis, for the set U' there is a balanced generalized list T-

coloring problem which does not have a good labeling. Now, by induction applied to this new problem, there is a balanced generalized list T-coloring problem for the set U which does not have a good labeling. $\square$

As an immediate corollary of Lemma 4.9, we obtain that if a balanced generalized list T-coloring problem on a complete graph does not have a good labeling, then all sets $t(e)$ must be arithmetic.

LEMMA 4.10. *Let (K_n, Λ, t) be a balanced generalized list T-coloring problem. If K_n has an edge e such that $t(e)$ is not arithmetic, then the problem (K_n, Λ, t) allows a good labeling.*

Proof. Let u and v be the end-vertices of the edge e such that $t(e)$ is not arithmetic. If the problem (K_n, Λ, t) does not have a good labeling, then apply Lemma 4.9 with $U = \{u, v\}$ to get a balanced generalized list T-coloring problem (K_2, Λ', t') with no good labeling such that $t'(e) = t(e)$ is not arithmetic. But this is impossible by Lemma 4.4. $\square$

We now focus on the relation between lists $\Lambda : V(G) \to 2^{\mathbb{N}}$ and forbidden sets $t : E(G) \to 2^{\mathbb{N}}$ in generalized list T-coloring problems on complete graphs with no good labelings.

LEMMA 4.11. *Let (K_n, Λ, t) be a balanced generalized list T-coloring problem. Let $v_1, \ldots, v_n$ be an arbitrary ordering of the vertices of K_n. If (K_n, Λ, t) does not have a good labeling, then there exist numbers $k_1 < k_2 < \cdots < k_{n-1}$ such that*

$$\Lambda(v_n) = \biguplus_{1 \le i \le n-1} (t(v_i v_n) + k_i).$$

Moreover, for each k_i, $i = 1, \ldots, n-1$, and each $j = i, \ldots, n$,

$$k_i = \min\left(\Lambda(v_j) \setminus \bigcup_{1 \le i' < i} (t(v_{i'} v_j) + k_{i'})\right).$$

In particular, $k_1 = \Lambda_{\min}$.

Proof. The proof proceeds by induction on n. The lemma vacuously holds for $n = 1$. For $n = 2$, the lemma follows from Lemma 4.4. Suppose now that $n \ge 3$ and set $k_1 = \Lambda_{\min}$. Let $(K_{n-1}, \Lambda', t') = (K_n, \Lambda, t)[v_1 {\to} \Lambda_{\min}]$. Note that the problem (K_{n-1}, Λ', t') is balanced, since otherwise the problem (K_n, Λ, t) would have a good labeling. By the induction hypothesis, there are numbers $k_2 < \cdots < k_{n-1}$ such that

$$\Lambda'(v_n) = \biguplus_{2 \le i \le n-1} (t'(v_i v_n) + k_i).$$

Moreover, for all $i \in \{2, \ldots, n-1\}$ and $j \in \{i, \ldots, n\}$, the following holds:

$$k_i = \min\left(\Lambda'(v_j) \setminus \bigcup_{2 \le i' < i} (t'(v_{i'} v_j) + k_{i'})\right).$$

Since $\Lambda'(v_j) = \Lambda(v_j) \setminus (t(v_1 v_j) + \Lambda_{\min})$ and $t(e) = t'(e)$ for each edge of K_{n-1}, we have

$$\Lambda(v_n) = \Lambda'(v_n) \uplus (t(v_1 v_n) + k_1) = \biguplus_{1 \le i \le n-1} (t(v_i v_n) + k_i).$$

Similarly, we have for all i with $2 \le i \le n-1$, and all j with $i \le j \le n$,

$$k_i = \min\left(\Lambda(v_j) \setminus \bigcup_{1 \le i' < i} (t(v_{i'}v_j) + k_{i'})\right).$$

The final equality, which follows for all $1 \le j \le n$ from the choice of k_1 and the fact that $\Lambda_{\min}$ is contained in all lists (by Lemma 4.1), is

$$k_1 = \min\left(\Lambda(v_j) \setminus \bigcup_{1 \le i' < 1} (t(v_{i'}v_j) + k_{i'})\right) = \min \Lambda(v_j) = \Lambda_{\min}. \qquad \square$$

Roughly speaking, if a balanced generalized list T-coloring problem (K_n, Λ, t) does not have a good labeling, all sets $t(e)$ incident with the same vertex must share the same difference, as stated in the next lemma.

LEMMA 4.12. *Let (K_n, Λ, t), $n \ge 4$, be a balanced generalized list T-coloring problem. If (K_n, Λ, t) has no good labeling, then all the sets $t(e)$, $e \in E(K_n)$ are arithmetic, and all the sets $t(e)$ for all edges e incident with the same vertex v share the same difference.*

Proof. It follows from Lemma 4.10 that all the sets $t(e)$ are arithmetic. By Lemma 4.9, it is enough to prove the claim for $n = 4$. Let us assume that $n = 4$ and v, x, y, and z are the vertices of K_4 such that the sets $t(e)$ for the edges e incident with the vertex v do not share the same difference. Let d_x, d_y, and d_z be the differences and k_x, k_y, and k_z the sizes of the arithmetic sets $t(vx)$, $t(vy)$, and $t(vz)$, respectively. By our assumption, at most one of the three numbers k_x, k_y, and k_z is equal to one. Hence, we may assume that $k_x \ge 1$, $k_y \ge 2$, and $k_z \ge 2$. Moreover, the three differences d_x, d_y, and d_z are not all the same by the choice of v. We distinguish three cases and eventually derive a contradiction in each of them:

- $k_x = 1$, $k_y \ge 2$, $k_z \ge 2$, *and* $d_y < d_z$ (the case $d_y > d_z$ is symmetric)
  Note that $t(vx) = \{0\}$. Consider the problem $(K_4, \Lambda, t)[x \to \Lambda_{\max}]$ obtained by assigning the color $\Lambda_{\max}$ to the vertex x. This is a balanced generalized list T-coloring problem which does not have a good labeling. Note that its underlying graph is a triangle. Recall that the sets of forbidden differences on its edges are $t(vy) = \mathrm{Ar}_{d_y}(k_y)$ and $t(vz) = \mathrm{Ar}_{d_z}(k_z)$. Hence, $t(yz) = \{0\}$ by Lemma 4.8. By Theorem 4.7, the vertex v must be of the third type since $d_y \ne d_z$. In addition, k_y and k_z satisfy $k_y d_y = k_z d_z = \mathrm{lcm}(d_y, d_z)$. In particular, d_z is not divisible by d_y (recall that $k_y \ge 2$ and $k_z \ge 2$). Since the vertex v is of the third type in $(K_4, \Lambda, t)[x \to \Lambda_{\max}]$, its list in the new problem $(K_4, \Lambda, t)[x \to \Lambda_{\max}]$ is equal to the following set:

  $$(\Lambda_{\min} + \mathrm{Ar}_{d_y}(k_y)) \cup (\Lambda_{\min} + \mathrm{Ar}_{d_z}(k_z)) \cup \{\Lambda_{\min} + \mathrm{lcm}(d_y, d_z)\}.$$

  Hence, we infer that

  $$\Lambda(v) = (\Lambda_{\min} + \mathrm{Ar}_{d_y}(k_y)) \cup (\Lambda_{\min} + \mathrm{Ar}_{d_z}(k_z)) \cup \{\Lambda_{\min} + \mathrm{lcm}(d_y, d_z), \Lambda_{\max}\}. \tag{4.3}$$

  Similarly, considering the problem $(K_4, \Lambda, t)[x \to \Lambda_{\min}]$ yields

  $$\Lambda(v) = (\Lambda_{\max} - \mathrm{Ar}_{d_y}(k_y)) \cup (\Lambda_{\max} - \mathrm{Ar}_{d_z}(k_z)) \cup \{\Lambda_{\max} - \mathrm{lcm}(d_y, d_z), \Lambda_{\min}\}. \tag{4.4}$$

The second largest element of $\Lambda(v)$ according to (4.3) is $\Lambda_{\min} + \mathrm{lcm}(d_y, d_z)$ and according to (4.4) is $\Lambda_{\max} - d_y$. Hence, $\Lambda_{\max} - \Lambda_{\min} = \mathrm{lcm}(d_y, d_z) + d_y$. On the other hand, the largest element of $\Lambda(v)$ which is not congruent to $\Lambda_{\max}$ modulo d_y is equal to $\Lambda_{\min} + \mathrm{lcm}(d_y, d_z) - d_z$ according to (4.3) and equal to $\Lambda_{\max} - d_z$ according to (4.4) (recall that $k_y d_y = k_z d_z = \mathrm{lcm}(d_y, d_z)$). Hence, we infer that $\Lambda_{\max} - \Lambda_{\min} = \mathrm{lcm}(d_y, d_z)$ contradicts our previously established equality $\Lambda_{\max} - \Lambda_{\min} = \mathrm{lcm}(d_y, d_z) + d_y$.

- $k_x \geq 2$, $k_y \geq 2$, $k_z \geq 2$, *and* $d_x < d_y < d_z$
The problem obtained by assigning the color $\Lambda_{\max}$ to the vertex z does not have a good labeling. In this new problem, the vertex v must be of the third type described in Theorem 4.7 because the differences of the sets of the edges incident with v are different. We infer that $k_x d_x = k_y d_y = \mathrm{lcm}(d_x, d_y)$. Since $k_x \geq 2$ and $k_y \geq 2$, d_y is not divisible by d_x. And by Lemma 4.8, it must be $t(xy) = \{0\}$. Symmetric arguments yield $k_x d_x = k_z d_z = \mathrm{lcm}(d_x, d_z)$, $k_y d_y = k_z d_z = \mathrm{lcm}(d_y, d_z)$, and $t(xz) = t(yz) = \{0\}$. Let l be the following number:

$$l = k_x d_x = k_y d_y = k_z d_z = \mathrm{lcm}(d_x, d_y) = \mathrm{lcm}(d_x, d_z) = \mathrm{lcm}(d_y, d_z). \tag{4.5}$$

Consider again the problem $(K_4, \Lambda, t)[z \to \Lambda_{\max}]$ obtained by assigning the color $\Lambda_{\max}$ to the vertex z. Since $t(xz) = t(yz) = \{0\}$, the color $\Lambda_{\min}$ remains in the lists of the vertices of x and y. Then the color $\Lambda_{\min}$ must remain also in the list of the vertex v by Lemma 4.1. Hence, the list of v in the obtained problem is equal to the following:

$$(\Lambda_{\min} + t(vx)) \cup (\Lambda_{\min} + t(vy)) \cup \{\Lambda_{\min} + l\}.$$

The following inclusion immediately follows:

$$(\Lambda_{\min} + \mathrm{Ar}_{d_x}(k_x)) \cup (\Lambda_{\min} + \mathrm{Ar}_{d_y}(k_y)) \cup \{\Lambda_{\min} + l\} \subseteq \Lambda(v).$$

By symmetry, we also have the following:

$$(\Lambda_{\min} + \mathrm{Ar}_{d_x}(k_x)) \cup (\Lambda_{\min} + \mathrm{Ar}_{d_z}(k_z)) \cup \{\Lambda_{\min} + l\} \subseteq \Lambda(v).$$

The size of the following set is $k_x + k_y + k_z - 1$ by (4.5):

$$(\Lambda_{\min} + \mathrm{Ar}_{d_x}(k_x)) \cup (\Lambda_{\min} + \mathrm{Ar}_{d_y}(k_y)) \cup (\Lambda_{\min} + \mathrm{Ar}_{d_z}(k_z)) \cup \{\Lambda_{\min} + l\}. \tag{4.6}$$

Since the problem (K_4, Λ, t) is balanced, the size of $\Lambda(v)$ is $k_x + k_y + k_z$. We now have (observe that the missing color in (4.6) can be only $\Lambda_{\max}$)

$$\Lambda(v) = (\Lambda_{\min} + \mathrm{Ar}_{d_x}(k_x)) \cup (\Lambda_{\min} + \mathrm{Ar}_{d_y}(k_y)) \cup (\Lambda_{\min} + \mathrm{Ar}_{d_z}(k_z)) \cup \{\Lambda_{\min} + l, \Lambda_{\max}\}. \tag{4.7}$$

A symmetric argument based on the problems obtained by assigning the color $\Lambda_{\min}$ to some of the vertices gives the following equality:

$$\Lambda(v) = (\Lambda_{\max} - \mathrm{Ar}_{d_x}(k_x)) \cup (\Lambda_{\max} - \mathrm{Ar}_{d_y}(k_y)) \cup (\Lambda_{\max} - \mathrm{Ar}_{d_z}(k_z)) \cup \{\Lambda_{\max} - l, \Lambda_{\min}\}. \tag{4.8}$$

Now, the equalities (4.7) and (4.8) are compared: the second largest element of $\Lambda(v)$ according to (4.7) is $\Lambda_{\min} + l$ and according to (4.8) is $\Lambda_{\max} - d_x$.

Hence, we can infer that $\Lambda_{\max} - \Lambda_{\min} = l + d_x$. The largest element of $\Lambda(v)$ which is not congruent to $\Lambda_{\max}$ modulo d_x is equal to $\Lambda_{\min} + l - d_y$ according to the equalities (4.5) and (4.7). But the largest element which is not congruent to $\Lambda_{\max}$ modulo d_x is equal to $\Lambda_{\max} - d_y$ according to the equalities (4.5) and (4.8). Hence, we have $\Lambda_{\max} - \Lambda_{\min} = l$, which contradicts $\Lambda_{\max} - \Lambda_{\min} = l + d_x$.

- *$k_x \geq 2$, $k_y \geq 2$, $k_z \geq 2$, and $d_x = d_y \neq d_z$*
As in the previous case, consider the problems $(K_4, \Lambda, t)[x \to \Lambda_{\max}]$ and $(K_4, \Lambda, t)[y \to \Lambda_{\max}]$ and conclude that $t(xz) = t(yz) = \{0\}$. In particular, it is possible to define $l = k_x d_x = k_y d_y = k_z d_z = \mathrm{lcm}(d_x, d_z)$ and $k_x = k_y$.
Consider again the problem $(K_4, \Lambda, t)[y \to \Lambda_{\max}]$. Since $t(yz) = \{0\}$, the color $\Lambda_{\min}$ remains in the list of the vertex z. Then the color $\Lambda_{\min}$ must also remain in the list of the vertex v by Lemma 4.1. Hence, the list of v in the new problem is equal to the following set:

$$(\Lambda_{\min} + t(vx)) \cup (\Lambda_{\min} + t(vz)) \cup \{\Lambda_{\min} + l\}.$$

The way in which the new problem was obtained immediately implies the following equality:

$$\begin{aligned}\Lambda(v) = ((\Lambda_{\min} + \mathrm{Ar}_{d_x}(k_x)) \cup (\Lambda_{\min} + \mathrm{Ar}_{d_z}(k_z)) \cup \{\Lambda_{\min} + l\})\\ \uplus (\Lambda_{\max} - \mathrm{Ar}_{d_y}(k_y)).\end{aligned} \tag{4.9}$$

A symmetric argument which is based on the problem obtained by assigning the color $\Lambda_{\min}$ to the vertex y gives the following:

$$\begin{aligned}\Lambda(v) = ((\Lambda_{\max} - \mathrm{Ar}_{d_x}(k_x)) \cup (\Lambda_{\max} - \mathrm{Ar}_{d_z}(k_z)) \cup \{\Lambda_{\max} - l\})\\ \uplus (\Lambda_{\min} + \mathrm{Ar}_{d_y}(k_y)).\end{aligned} \tag{4.10}$$

The equalities $k_x = k_y$ and $d_x = d_y$ imply that $\Lambda_{\min} + \mathrm{Ar}_{d_x}(k_x) = \Lambda_{\min} + \mathrm{Ar}_{d_y}(k_y)$ and $\Lambda_{\max} - \mathrm{Ar}_{d_x}(k_x) = \Lambda_{\max} - \mathrm{Ar}_{d_y}(k_y)$. This combined with the equalities (4.9) and (4.10) yields the following:

$$(\Lambda_{\min}+d_z+\mathrm{Ar}_{d_z}(k_z-1)) \uplus \{\Lambda_{\min}+l\} = (\Lambda_{\max}-d_z-\mathrm{Ar}_{d_z}(k_z-1)) \uplus \{\Lambda_{\max}-l\}. \tag{4.11}$$

Since $l = k_z d_z$, we can simplify (4.11) to the following equality:

$$\mathrm{Ar}_{d_z}(k_z) + \Lambda_{\min} + d_z = \Lambda_{\max} - d_z - \mathrm{Ar}_{d_z}(k_z).$$

Hence, we can infer (by considering the largest and the smallest element in the sets above) that $\Lambda_{\max} - \Lambda_{\min} = l + d_z$.
Let us consider now the problem $(K_3, \Lambda', t') = (K_4, \Lambda, t)[z \to \Lambda_{\max}]$. First, we have by the equality (4.10) (the union in the next equality is disjoint because the new problem must be balanced)

$$\Lambda'(v) = \Lambda(v) \setminus (\Lambda_{\max} - t(vz)) = \Lambda(v) \setminus (\Lambda_{\max} - \mathrm{Ar}_{d_z}(k_z))$$

$$= (\Lambda_{\min} + \mathrm{Ar}_{d_x}(k_x)) \uplus (\Lambda_{\max} - d_x - \mathrm{Ar}_{d_x}(k_x)).$$

Observe that, by Lemma 4.9, the problem (K_3, Λ', t') is a balanced generalized list T-coloring problem which does not have a good labeling. Let $\Lambda'_{\min}$ and

$\Lambda'_{\max}$ be the smallest and the largest element contained in the lists Λ'. Since $\Lambda'_{\min} = \Lambda_{\min}$ and $\Lambda'_{\max} = \max\{\Lambda_{\min} + l - d_x, \Lambda_{\max} - d_x\} = \Lambda_{\max} - d_x = \Lambda_{\min} + l + d_z - d_x$ are not congruent modulo d_x, all the vertices v, x, and y in the problem (K_3, Λ', t') must be of the first type described in the statement of Theorem 4.7. Hence, we infer that $t(xy) = t'(xy) = \mathrm{Ar}_{d_x}(k_x)$ and

$$\Lambda'(x) = \Lambda'(y) = \Lambda'(v) = (\Lambda_{\min} + \mathrm{Ar}_{d_x}(k_x)) \uplus (\Lambda_{\max} - d_x - \mathrm{Ar}_{d_x}(k_x)).$$

In particular,

$$(\Lambda_{\min} + \mathrm{Ar}_{d_x}(k_x)) \uplus (\Lambda_{\max} - d_x - \mathrm{Ar}_{d_x}(k_x)) \subseteq \Lambda(x). \tag{4.12}$$

A symmetric argument based on the problem obtained by assigning the color $\Lambda_{\min}$ to the vertex z yields the following inclusion:

$$(\Lambda_{\max} - \mathrm{Ar}_{d_x}(k_x)) \uplus (\Lambda_{\min} + d_x + \mathrm{Ar}_{d_x}(k_x)) \subseteq \Lambda(x). \tag{4.13}$$

By comparing the inclusions (4.12) and (4.13), we get the following:

$$(\Lambda_{\min} + \mathrm{Ar}_{d_x}(k_x + 1)) \uplus (\Lambda_{\max} - \mathrm{Ar}_{d_x}(k_x + 1)) \subseteq \Lambda(x).$$

Thus the size of $\Lambda(x)$ must be at least $2k_x + 2$. On the other hand, the t-degree of x in the problem (K_n, Λ, t) is $|t(xy)| + |t(xz)| + |t(xv)| = 2k_x + 1$. This contradicts the assumption that the problem (K_n, Λ, t) is balanced. □

Now, we extend the argument from the previous lemma and show that all the sets $t(e)$ must share the same difference. Note that we cannot derive this conclusion immediately from Lemma 4.12 since there could exist edges e with $t(e) = \{0\}$.

LEMMA 4.13. *Let (K_n, Λ, t) be a balanced generalized list T-coloring problem. If (K_n, Λ, t) has no good labeling and $n \geq 4$, then all the sets $t(e)$ for $e \in E(K_n)$ are arithmetic sets with the same difference.*

Proof. By Lemma 4.12, all the sets $t(e)$ are arithmetic and the sets $t(e)$ for edges e incident with the same vertex have the same difference. If there are edges e and e' with $|t(e)|, |t(e')| \geq 2$ such that $t(e)$ and $t(e')$ do not have the same difference, then the edges e and e' cannot be incident. Let $e = vw$ and $e' = xy$. By Lemma 4.9, it is enough now to prove the statement for $n = 4$, i.e., a balanced generalized list T-coloring problem whose underlying graph is the complete graph of order four comprised of the vertices v, w, x, and y. By Lemma 4.12, we have $t(vx) = t(wx) = t(vy) = t(wy) = \{0\}$. Let k_{vw} and k_{xy} be the sizes of the sets $t(vw)$ and $t(xy)$, respectively. Similarly, let d_{vw} and d_{xy} be their differences. Recall that we have assumed that $d_{vw} \neq d_{xy}$.

Consider the problem $(G', \Lambda', t') = (K_n, \Lambda, t)[y \to \Lambda_{\max}]$. The problem (G', Λ', t') is balanced and it does not have a good labeling. Theorem 4.7 implies the following equalities:

$$\begin{aligned} &\Lambda'(v) = \Lambda'(w) = \Lambda_{\min} + \mathrm{Ar}_{d_{vw}}(k_{vw} + 1) \text{ and} \\ &\Lambda'(x) = \{\Lambda_{\min}, \Lambda_{\min} + d_{vw}k_{vw}\}. \end{aligned} \tag{4.14}$$

Hence, $\Lambda_{\min} + \mathrm{Ar}_{d_{vw}}(k_{vw} + 1) \subseteq \Lambda(v)$. Next, consider the problem $(K_n, \Lambda, t)[y \to \Lambda_{\min}]$. By a similar argument as before, we obtain that $\Lambda_{\max} - \mathrm{Ar}_{d_{vw}}(k_{vw} + 1) \subseteq \Lambda(v)$. Since $|\Lambda(v)| = \deg_t(v) = k_{vw} + 2$, we can infer that $\Lambda_{\max} - \Lambda_{\min} = d_{vw}(k_{vw} + 1)$ and $\Lambda(v) = \Lambda_{\min} + \mathrm{Ar}_{d_{vw}}(k_{vw} + 2)$. Similarly, we may determine that the lists of the vertices w, x, and y are as follows:

$$\begin{aligned} \Lambda(v) &= \Lambda(w) = \Lambda_{\min} + \mathrm{Ar}_{d_{vw}}(k_{vw} + 2) \quad \text{and} \\ \Lambda(x) &= \Lambda(y) = \Lambda_{\min} + \mathrm{Ar}_{d_{xy}}(k_{xy} + 2). \end{aligned}$$

But we know that $\Lambda_{\min} + d_{vw}k_{vw} = \Lambda_{\max} - d_{vw} \in \Lambda(x)$ by (4.14). Since $\Lambda(x) = \Lambda_{\min} + \mathrm{Ar}_{d_{xy}}(k_{xy}+2)$, all the elements of $\Lambda(x)$ are congruent with $\Lambda_{\max}$ modulo d_{xy}. In particular, $\Lambda_{\max} - d_{vw}$ and $\Lambda_{\max}$ are congruent modulo d_{xy}. We infer that $d_{xy} \mid d_{vw}$. By symmetry, we also infer that $d_{vw} \mid d_{xy}$. Hence, we conclude $d_{xy} = d_{vw}$—a contradiction. ∎

Finally, we extend our arguments to get some properties of the lists in balanced generalized list T-coloring problems (K_n, Λ, t) with no good labeling.

Lemma 4.14. *Let (K_n, Λ, t) be a balanced generalized list T-coloring problem with $n \geq 3$, which does not have a good labeling, and let v be a vertex of the graph K_n. If all the sets $t(e)$ for edges e incident with the vertex v are arithmetic with the same difference d but there exist two edges $e, e' \in E(K_n)$ incident with v for which $t(e) \neq t(e')$, then all the elements of the list $\Lambda(v)$ are congruent modulo d.*

Proof. We prove by induction on n that if the elements of $\Lambda(v)$ are not congruent modulo d, then the problem (K_n, Λ, t) has a good labeling. This will establish the claim of the lemma. If $n = 3$, this is true by Theorem 4.7 because the vertex v must be of the second type.

Suppose now that $n \geq 4$. Let $k_{\min}$ and $k_{\max}$ be the minimum and the maximum size of the lists $t(e)$ for edges e incident with the vertex v. By the assumptions of the lemma, $k_{\min} < k_{\max}$. Let $v_{\min}$ and $v_{\max}$ be vertices of G such that $t(vv_{\min}) = \mathrm{Ar}_d(k_{\min})$ and $t(vv_{\max}) = \mathrm{Ar}_d(k_{\max})$. By Lemma 4.1, we can also assume that the colors $\Lambda_{\min}$ and $\Lambda_{\max}$ are contained in the lists of all the vertices. We consider three cases:

- If *the number of elements of $\Lambda(v)$ congruent with $\Lambda_{\min}$ modulo d is smaller than $k_{\max}$*, then $\Lambda_{\min} + t(vv_{\max}) = \Lambda_{\min} + \mathrm{Ar}_d(k_{\max}) \not\subseteq \Lambda(v)$. Hence, the problem $(K_n, \Lambda, t)[v_{\max} \to \Lambda_{\min}]$ is overbalanced. The problem (K_n, Λ, t) has then a good labeling by Lemma 2.1 and Theorem 3.1.
- If *the number of elements of $\Lambda(v)$ congruent with $\Lambda_{\min}$ modulo d is greater than $k_{\max}$*, we proceed as follows: let u be a vertex distinct from v, $v_{\min}$, and $v_{\max}$. The problem $(K_n, \Lambda, t)[u \to \Lambda_{\min}]$ is overbalanced or the list of the vertex v contain two elements which are not congruent modulo d. In the former case, it has a good labeling by Theorem 3.1. In the latter case, it has a good labeling by induction. Hence, the problem (K_n, Λ, t) has a good labeling by Lemma 2.1.
- The final case is that *the number of elements of $\Lambda(v)$ congruent with $\Lambda_{\min}$ modulo d is exactly $k_{\max}$*. If there is a vertex $u \neq v_{\min}$ with $|t(vu)| < k_{\max}$, then $(K_n, \Lambda, t)[u \to \Lambda_{\min}]$ is overbalanced or the list of the vertex v contains two elements which are not congruent modulo d. Similarly, as in the previous case, we conclude that the problem (K_n, Λ, t) has a good labeling. The other possibility is that for each vertex $u \neq v_{\min}$, we have $t(vu) = t(vv_{\max}) = \mathrm{Ar}_d(k_{\max})$. Consider now the problem $(K_{n-1}, \Lambda', t') = (K_n, \Lambda, t)[v_{\min} \to \Lambda_{\min}]$. Since the problem (K_n, Λ, t) is assumed not to have a good labeling, the problem (K_{n-1}, Λ', t') should admit no good labeling as well. In particular, the problem (K_{n-1}, Λ', t') is balanced. By Lemma 4.11, the number of elements with the same remainder modulo d contained in the set $\Lambda'(v)$ is divisible by $k_{\max}$ because $t'(e) = \mathrm{Ar}_d(k_{\max})$ for every edge e incident with v. But the set $\Lambda'(v)$ contains exactly $k_{\max} - k_{\min} < k_{\max}$ elements congruent modulo d with $\Lambda_{\min}$ (of the original problem (K_n, Λ, t)). ∎

We may now extend the arguments of Lemma 4.14 to show that all elements of the lists are congruent modulo d, where d is the common difference of all the sets $t(e)$.

Lemma 4.15. *Let (K_n, Λ, t) be a balanced generalized list T-coloring problem which does not have a good labeling. If all $t(e)$ for $e \in E(K_n)$ are arithmetic sets with the same difference d and there exist edges $e, e' \in E(K_n)$ such that $t(e) \neq t(e')$, then all the elements of the union $\bigcup_{v\in V(K_n)} \Lambda(v)$ are congruent modulo d.*

Proof. By the assumption of the lemma, there is a vertex w of K_n which satisfies the assumption of Lemma 4.14. Hence, the elements of the list $\Lambda(w)$ are congruent modulo d. Let w' be a vertex distinct from w. Order vertices of K_n in the sequence $v_1, v_2, \dots, v_n$ in such a way that $v_{n-1} = w$ and $v_n = w'$. By Lemma 4.11, there exist numbers $k_1 < k_2 < \cdots < k_{n-1}$ such that

$$\Lambda(w') = \biguplus_{1\le i\le n-1} (t(w'v_i) + k_i).$$

In addition, the following holds for each i, $1 \le i \le n-1$:

$$k_i = \min\left(\Lambda(w) \setminus \bigcup_{1\le i'<i} (t(wv_i) + k_i)\right).$$

In particular, $k_i \in \Lambda(w)$ and since all the sets $t(w'v_i)$ have the same difference d, all the elements of the list $\Lambda(w')$ are congruent with all the elements of $\Lambda(w)$ modulo d. Since the choice of w' was arbitrary, the proof is completed. □

In the proof of the main theorem of this subsection, we use the Brooks-type theorem for the channel assignment problem on complete graphs from [10]. We formulate it in our notation.

Theorem 4.16. *Let (K_n, Λ, t) be a balanced generalized list T-coloring problem such that each list $t(e)$ for $e \in E(K_n)$ is an arithmetic set with difference 1. Let $V(K_n) = \{v_1, \dots, v_n\}$. The problem (K_n, Λ, t) does not have a good labeling if and only if one of the following holds:*

- *There exist integers $1 \le a$ and $0 \le k_1 < \cdots < k_{n-1}$ such that*
  - $k_i + a \le k_{i+1}$ *for each* $i = 1, \dots, n-2$,
  - $t(e) = \mathrm{Ar}_1(a)$ *for each edge* $e \in E(K_n)$, *and*
  - $\Lambda(v_i) = \bigcup_{1\le j\le n-1}(k_j + \mathrm{Ar}_1(a))$ *for each vertex* v_i *of* K_n.
- *There exist integers $1 \le a < b$ and $0 \le k$ such that (possibly after an appropriate permutation of the vertices)*
  - $t(e) = \begin{cases} \mathrm{Ar}_1(a) & \textit{if } e \textit{ is incident with the vertex } v_n,\\ \mathrm{Ar}_1(b) & \textit{otherwise.}\end{cases}$
  - $\Lambda(v_i) = \begin{cases} k + \mathrm{Ar}_1(a + b(n-2)) & \textit{if } i \neq n,\\ \bigcup_{0\le j\le n-2}(k + bj + \mathrm{Ar}_1(a)) & \textit{otherwise.}\end{cases}$

We can now characterize in a similar way balanced generalized list T-coloring problems whose underlying graph is a complete graph and which do not have a good labeling (an example of such a balanced generalized list T-coloring problem with no good labeling can be found in Figure 4.2).

Theorem 4.17. *Let (K_n, Λ, t) be a balanced generalized list T-coloring problem with $n \ge 4$. Let $V(K_n) = \{v_1, \dots, v_n\}$. The problem (K_n, Λ, t) does not have a good labeling if and only if it is one of the following two types:*

- *There exist integers $1 \le a$, $1 \le d$, and $0 \le k_1 < \cdots < k_{n-1}$ such that*
  - $t(e) = \mathrm{Ar}_d(a)$ *for all* $e \in E(K_n)$ *and*
  - $\Lambda(v_i) = \biguplus_{1\le j\le n-1}(k_j + \mathrm{Ar}_d(a))$ *for all* $1 \le i \le n$.
- *There exist integers $1 \le a < b$, $1 \le d$, and $0 \le k$ such that (possibly after an appropriate permutation of the vertices)*

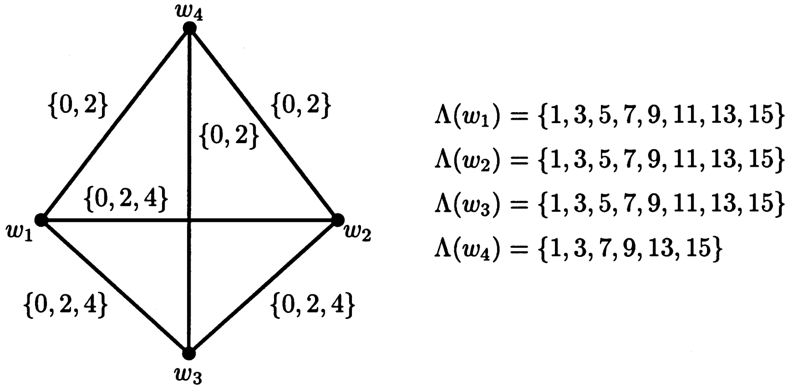

w_4
$\{0,2\}$
$\{0,2\}$
$\{0,2\}$
$\{0,2,4\}$
w_1
w_2
$\{0,2,4\}$
$\{0,2,4\}$
w_3
$\Lambda(w_1) = \{1,3,5,7,9,11,13,15\}$
$\Lambda(w_2) = \{1,3,5,7,9,11,13,15\}$
$\Lambda(w_3) = \{1,3,5,7,9,11,13,15\}$
$\Lambda(w_4) = \{1,3,7,9,13,15\}$

FIG. 4.2. *An example of a balanced generalized list T-coloring problem with an underlying graph being K_4. The sets of forbidden differences are at the centers of the corresponding edges and the lists of colors for the vertices are on the right.*

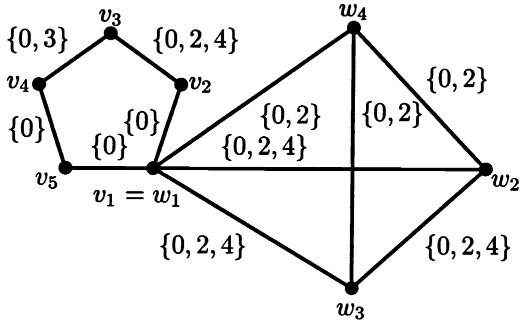
- $t(e) = \begin{cases} \mathrm{Ar}_d(a) & \textit{if } e \textit{ is incident with the vertex } v_n, \\ \mathrm{Ar}_d(b) & \textit{otherwise.} \end{cases}$
- $\Lambda(v_i) = \begin{cases} k + \mathrm{Ar}_d(a + b(n-2)) & \textit{if } i \neq n, \\ \bigcup_{0 \le j \le n-2} (k + bjd + \mathrm{Ar}_d(a)) & \textit{otherwise.} \end{cases}$

Proof. It is easy to check that if a problem (K_n, Λ, t) is of one of the above two types, then it is balanced. If it is of the first type described above, then in any labeling from the lists, at most one vertex of K_n has a color from $k_j + \mathrm{Ar}_d(a)$. By the pigeon-hole principle, the problem (K_n, Λ, t) cannot have a good labeling. If the problem (K_n, Λ, t) is of the second type described above, we can assume without loss of generality that $d \mid k$. Observe that the problem (K_n, Λ, t) has a good labeling if and only if the problem (K_n, Λ', t') with the parameters $a' = a$, $b' = b$, $d' = 1$, and $k' = k/d$ has a good labeling. But this problem has no good labeling by Theorem 4.16.

We show that each balanced generalized list T-coloring problem (K_n, Λ, t) with no good labeling is of one of the two types described in the statement of the theorem. By Lemma 4.10, for each $e \in E(G)$ the set $t(e)$ is arithmetic. By Lemma 4.13, all the sets $t(e)$, $e \in E(G)$ have the same difference d. If there are edges e and e' such that $t(e) \neq t(e')$, then all the elements of all the lists $\Lambda(v)$, $v \in V(K_n)$ are congruent modulo d by Lemma 4.15. We may assume that $d \mid \Lambda_{\min}$, i.e., all the elements of all the lists $\Lambda(v)$ are divisible by d. Consider now the balanced problem (K_n, Λ', t') with $\Lambda'(v) = \{\frac{k}{d} \mid k \in \Lambda(v)\}$ and $t'(e) = \{\frac{k}{d} \mid k \in t(e)\}$. Observe that the problem (K_n, Λ, t) has a good labeling if and only if the problem (K_n, Λ', t') has a good labeling. Then, by the assumption, the problem (K_n, Λ', t') does not have a good labeling. Since the common difference of all the sets $t'(e)$ is one and there are edges e and e' such that $t'(e) \neq t'(e')$, it must be of the second type described in Theorem 4.16. Let a', b', and k' be the parameters from the statement of Theorem 4.16. We may conclude that the problem (K_n, Λ, t) is of the second type with the parameters $a = a'$, $b = b'$, and $k = k'd$.

The remaining case is that all the sets $t(e)$ for $e \in E(G)$ are the same. Suppose that they are equal to $\mathrm{Ar}_d(k)$. By Lemma 4.11, there exist integers $k_1 < \cdots < k_{n-1}$ such that for all the vertices v of K_n,

$$\Lambda(v) = \biguplus_{1 \le i \le n-1} (\mathrm{Ar}_d(k) + k_i).$$



$$\Lambda(v_1) = \{1,2,3,5,7,8,9,11,13,15\}$$
$$\Lambda(v_2) = \{2,4,6,8\}$$
$$\Lambda(v_3) = \{2,4,5,6,8\}$$
$$\Lambda(v_4) = \{2,5,8\}$$
$$\Lambda(v_5) = \{2,8\}$$
$$\Lambda(w_2) = \{1,3,5,7,9,11,13,15\}$$
$$\Lambda(w_3) = \{1,3,5,7,9,11,13,15\}$$
$$\Lambda(w_4) = \{1,3,7,9,13,15\}$$

FIG. 5.1. *An example of a balanced generalized list T-coloring problem with no good labeling such that the underlying graph of the problem is not* 2*-connected. The problem is obtained by gluing the problems depicted in Figures* 4.1 *and* 4.2.

Hence, the problem is of the first type described in the statement of this theorem. This completes the proof of the theorem. □

5. The general case. We show that Lemma 4.4 and Theorems 4.7 and 4.17 can be combined to provide a full characterization of all balanced generalized list T-coloring problems which do not have a good labeling (an example of a balanced generalized list T-coloring problem with no good labeling whose underlying graph is not 2-connected can be found in Figure 5.1).

THEOREM 5.1. *Let (G, Λ, t) be a balanced generalized list T-coloring problem where G is a connected graph and let $B_1, \dots, B_l$ be the blocks of G. The problem (G, Λ, t) does not have a good labeling if and only if there exists $\Lambda_i : V(B_i) \to 2^{\mathbb{N}}$ and $t_i : E(B_i) \to 2^{\mathbb{N}}$, $1 \le i \le l$, such that*

1. *it holds $\Lambda(v) = \biguplus_{\substack{1\le i\le l\\ v\in V(B_i)}} \Lambda_i(v)$ for each $v \in V(G)$,*
2. *$t(e) = t_i(e)$ for the unique index i satisfying $e \in E(B_i)$, and*
3. *each generalized list T-coloring problem (B_i, Λ_i, t_i) is balanced and does not have a good labeling.*

In particular, if (G, Λ, t) has no good labeling, then G is a Gallai tree and each (B_i, Λ_i, t_i) is as described in Lemma 4.4 *and in Theorems* 4.7 *and* 4.17.

Proof. We first prove that a balanced generalized list T-coloring problem (G, Λ, t) of the type described in the statement does not have a good labeling. The proof is by induction on the number l of the blocks. If $l = 1$, the statement straightforwardly follows from Lemma 4.4 and Theorems 4.7 and 4.17. Otherwise, let B_l be an end-block of the graph G. Let v be the cut-vertex contained in B_l. Assume for the sake of contradiction that there is a good labeling c for the problem (G, Λ, t). If $c(v) \in \Lambda_l(v)$, then c restricted to B_l is a good labeling for the problem (B_l, Λ_l, t_l), which is impossible. If $c(v) \notin \Lambda_l(v)$, then c is a good labeling for the balanced problem (G', Λ', t'):

$$V(G') = \bigcup_{1\le i<l} V(B_i),$$

$$E(G') = \bigcup_{1\le i<l} E(B_i),$$

$$\Lambda'(v) = \bigcup_{1\le i<l, v\in V(B_i)} \Lambda_i(v), \text{ and}$$
$$t'(e) = t_i(e) \text{ for the unique } i \text{ such that } e \in E(B_i).$$

But this is impossible by the assumption of the induction.

We now prove that if a problem (G, Λ, t) does not have a good labeling, then it is of the type described in the statement of the theorem. The proof again proceeds by induction on the number l of the blocks of G. If $l = 1$, the statement easily follows from Lemma 4.4 and Theorems 4.7 and 4.17.

Assume that $l \geq 2$. Let B_1 be an end-block of the graph G, let v be the cut-vertex contained in B_1, and let G' be the graph comprised by the blocks $B_2, \ldots, B_l$. Let Λ_1 and Λ' be the function Λ restricted to $V(B_1)$ and $V(G')$, respectively. Similarly, let t_1 and t' be the function t restricted to $E(B_1)$ and $E(G')$, respectively. Let L_1 be the set of all the colors $k \in \Lambda(v) = \Lambda_1(v)$ such that there is not a good labeling c for the problem (B_1, Λ_1, t_1) with $c(v) = k$. By Theorem 3.1, $|L_1| \leq \deg_{t_1}(v)$. Let L' be the set of all the colors $k \in \Lambda(v) = \Lambda'(v)$ such that there is not a good labeling c for the problem (G', Λ', t') with $c(v) = k$. By Theorem 3.1, $|L'| \leq \deg_{t'}(v)$. If $|L_1 \cup L'| < \deg_{t_1}(v) + \deg_{t'}(v) = \deg_t(v)$, then there is a good labeling c for the problem (G, Λ, t) such that $c(v) = k$, where $k \in \Lambda(v) \setminus (L_1 \cup L')$. Otherwise, $|L_1| = \deg_{t_1}(v)$, $|L'| = \deg_{t'}(v)$, and thus $\Lambda(v) = L_1 \uplus L'$. Reset $\Lambda_1(v) = L_1$ and $\Lambda'(v) = L'$. By the induction hypothesis, both problems (B_1, Λ_1, t_1) and (G', Λ', t') are of the type described in the statement of the theorem. Hence, it easily follows that the problem (G, Λ, t) is also of the desired type. □

It is straightforward to check that all the proofs in this paper are algorithmic and hence we may conclude with the following.

Corollary 5.2. *There is a polynomial-time algorithm which for each overbalanced generalized list T-coloring problem finds a good labeling. There is also a polynomial-time algorithm which for each balanced generalized list T-coloring problem decides whether the problem has a good labeling and, if so, the algorithm finds such a labeling.*

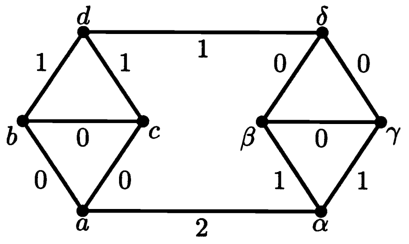
6. Conclusion.

Throughout the paper, all considered generalized list T-coloring problems (G, Λ, t) satisfy that $0 \in t(e)$ for all sets of forbidden differences (as a part of the definition of the generalized list T-coloring). A natural question to ask is what happens if we dismiss this requirement. In particular, the following problem naturally arises.

Problem 6.1. *Which (over)balanced generalized list T-coloring problems (G, Λ, t) do not have a good labeling when we do not require that $0 \in t(e)$ for all $e \in E(G)$?*

Surprisingly, it is *not* true that each such overbalanced generalized list T-coloring problem (G, Λ, t), where G is a connected graph, has a good labeling (this contrasts with the statement of Theorem 3.1 for overbalanced generalized list T-coloring problems with the requirement $0 \in t(e)$ for each set of forbidden differences). The example in Figure 6.1, which was derived in discussions of the second author and Jiří Sgall, shows that such a statement is not true. Moreover, this example has some interesting properties, such as its underlying graph is 2-connected but it is neither a cycle nor a complete graph, each set of forbidden differences is of size one, all the lists of vertices are the same except for a single vertex, etc.

Proposition 6.2. *If we dismiss a requirement that $0 \in t(e)$, then there exists an overbalanced generalized list T-coloring problem (G, Λ, t) which does not have a good*



$$\Lambda(a) = \Lambda(b) = \Lambda(c) = \Lambda(d) = \{1,2,3\}$$
$$\Lambda(\alpha) = \{1,2,3,4\}$$
$$\Lambda(\beta) = \Lambda(\gamma) = \Lambda(\delta) = \{1,2,3\}$$

FIG. 6.1. *An example of an overbalanced generalized list T-coloring problem (G, Λ, t) which does not have a good labeling if we dismiss the condition $0 \in t(e)$. Each edge has a single forbidden difference which is represented by the number at the middle of the edge. The lists $\Lambda(v)$, $v \in V(G)$, are described in the right part of the figure.*

labeling and which, in addition, satisfies

- *G is a 2-connected cubic graph.*
- *Each $\Lambda(v)$, $v \in V(G)$ is equal to $\{1,2,3\}$ except for a single vertex whose list is $\{1,2,3,4\}$.*
- *Each $t(e)$, $e \in E(G)$ is either $\{0\}$, $\{1\}$, or $\{2\}$. In particular, $|t(e)| = 1$ for every edge $e \in E(G)$.*

Proof. Consider the problem (G, Λ, t) depicted in Figure 6.1. It is easy to see that the problem has the properties from the statement of the proposition except that it does not have a good labeling. We now show that the problem (G, Λ, t) does not have a good labeling.

Assume for the sake of contradiction that the problem (G, Λ, t) has a good labeling λ. Let us consider first the case that $\lambda(b) = 2$. Then $\lambda(d)$ cannot be 1 or 3 because of the edge bd. Since $\lambda(c)$ is either 1 or 3 (the edge bc), $\lambda(d)$ cannot be 2 (the edge cd) either. But then the labeling λ cannot be proper. Hence, we can conclude that $\lambda(b) \neq 2$. Let $\lambda(c) \neq 2$ without loss of generality. We can now infer that $\lambda(a) = 2$ (consider the triangle abc) and $\lambda(b), \lambda(c) \in \{1,3\}$. By symmetry, it can actually be assumed that $\lambda(b) = 1$ and $\lambda(c) = 3$. Finally, we derive that $\lambda(d)$ is 1 or 3 (consider the edges bd and cd).

Since $\lambda(d) \in \{1,3\}$, the vertex δ cannot be assigned by the labeling λ the number 2, i.e., $\lambda(\delta) \neq 2$. By symmetry, we can assume that $\lambda(\beta) = 2$ (consider the triangle $\beta\gamma\delta$). Thus, $\lambda(\gamma)$ is equal to 1 or 3. If $\lambda(\gamma) = 3$, then $\lambda(\alpha)$ cannot be 1 or 3 because of the edge $\alpha\beta$, and it cannot be 2 or 4 because of the edge $\gamma\delta$. Thus, $\lambda(\gamma) = 1$ and $\lambda(\alpha) = 4$. We eventually obtain the contradiction since $\lambda(a) = 2$, $\lambda(\alpha) = 4$, and $t(a\alpha) = \{2\}$. □

We remark that it is not hard to show that the decision problem of whether an overbalanced generalized list T-coloring problem has a good labeling is NP-complete when we dismiss the requirement $0 \in t(e)$ for all edges e. This contrasts the fact that the corresponding problem for overbalanced generalized list T-coloring problems with this requirement is trivial (the answer is simply always "yes" if the underlying graph is connected), and even the corresponding problem for balanced generalized list T-coloring problems can be solved in polynomial time (Corollary 5.2).

Acknowledgments. The research was partly conducted while the second author was visiting the third author at Pacific Institute for the Mathematical Sciences (PIMS) at Simon Fraser University (SFU) in 2002. The authors would like to thank professor Pavol Hell for making their stay at PIMS and SFU pleasant and fruitful. The authors would also like to thank Jiří Sgall for his comments on the generalized list T-coloring without the requirement $0 \in t(e)$.

REFERENCES

[1] N. Alon and A. Zaks, *T-choosability in graphs*, Discrete Appl. Math., 82 (1998), pp. 1–13.

[2] O. V. Borodin, *Criterion of chromaticity of a degree prescription*, in Abstracts of IV All-Union Conf. on Theoretical Cybernetics, Novosibirsk, 1977, pp. 127–128 (in Russian).

[3] O. V. Borodin, *Problems of colouring and of covering the vertex set of a graph by induced subgraphs*, Ph.D. thesis, Novosibirsk State University, Novosibirsk, 1979 (in Russian).

[4] R. L. Brooks, *On colouring the nodes of a network*, Proc. Cambridge Phil. Soc., 37 (1941), pp. 194–197.

[5] G. J. Chang and D. Kuo, *The $L(2,1)$-labeling problem on graphs*, SIAM J. Discrete Math., 9 (1996), pp. 309–316.

[6] P. Erdős, A. L. Rubin, and H. Taylor, *Choosability in graphs*, in Proceedings of the West Coast Conference on Combinatorics, Graph Theory and Computing, Congr. Numer. XXVI, Utilitas Math., Winnipeg, MB, Canada, 1980, pp. 125–157.

[7] J. R. Griggs and R. K. Yeh, *Labelling graphs with a condition at distance* 2, SIAM J. Discrete Math., 5 (1992), pp. 586–595.

[8] W. K. Hale, *Frequency assignment: Theory and applications*, Proc. IEEE, 68 (1980), pp. 1497–1514.

[9] A. V. Kostochka, M. Stiebitz, and B. Wirth, *The colour theorems of Brooks and Gallai extended*, Discrete Math., 162 (1996), pp. 299–303.

[10] D. Král' and R. Škrekovski, *A theorem about the channel assignment problem*, SIAM J. Discrete Math., 16 (2003), pp. 426–437.

[11] J. Kratochvíl, Z. Tuza, and M. Voigt, *Brooks-type theorems for choosability with separation*, J. Graph Theory, 27 (1998), pp. 43–49.

[12] J. Kratochvíl, Z. Tuza, and M. Voigt, *New trends in the theory of graph colorings: Choosability and list coloring*, in Contemporary Trends in Discrete Mathematics, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 49, AMS, Providence, RI, 1999, pp. 183–197.

[13] D. D.-F. Liu, *T-colorings of graphs*, Discrete Math., 101 (1992), pp. 203–212.

[14] L. Lovász, *Three short proofs in graph theory*, J. Combin. Theory Ser. B, 19 (1975), pp. 269–271.

[15] C. McDiarmid, *Discrete mathematics and radio channel assignment*, in Recent Advances in Theoretical and Applied Discrete Mathematics, C. Linhares-Salas and B. Reed, eds., Springer-Verlag, New York, 2001.

[16] C. McDiarmid, *On the span in channel assignment problems: bounds, computing and counting*, Discrete Math., 266 (2003), pp. 387–397.

[17] M. Molloy and B. Reed, *Graph colouring and the Probabilistic Method*, Algorithms Combin. 23, Springer-Verlag, New York, 2001.

[18] F. Roberts, *T-colorings of graphs: Recent results and open problems*, Discrete Math., 93 (1991), pp. 229–245.

[19] D. Sakai, *Labeling chordal graphs: Distance two condition*, SIAM J. Discrete Math., 7 (1994), pp. 133–140.

[20] Z. Tuza, *Graph colorings with local constraints—a survey*, Discuss. Math. Graph Theory, 17 (1997), pp. 161–228.

[21] V. G. Vizing, *Colouring the vertices of a graph with prescribed colours*, Metody Diskretnogo Analiza Teorii Kodov i Skhem, 29 (1976), pp. 3–10 (in Russian).

[22] A. Waller, *An upper bound for list T-colourings*, Bull. London Math. Soc., 28 (1996), pp. 337–342.

[23] A. Waller, *Some results on list T-colourings*, Discrete Math., 174 (1997), pp. 357–363.

SIAM J. Discrete Math.
Vol. 19, No. 3, pp. 610–621

MULTILEVEL DISTANCE LABELINGS FOR PATHS AND CYCLES*

DAPHNE DER-FEN LIU† AND XUDING ZHU‡

Abstract. For a graph G, let $\operatorname{diam}(G)$ denote the diameter of G. For any two vertices u and v in G, let $d(u, v)$ denote the distance between u and v. A multilevel distance labeling (or distance labeling) for G is a function f that assigns to each vertex of G a nonnegative integer such that for any vertices u and v, $|f(u) - f(v)| \geq \operatorname{diam}(G) - d_G(u, v) + 1$. The span of f is the largest number in $f(V)$. The radio number of G, denoted by $rn(G)$, is the minimum span of a distance labeling for G. In this paper, we completely determine the radio numbers for paths and cycles.

Key words. channel assignment, distance two labeling, radio labeling, diameter

AMS subject classification. 05C15

DOI. 10.1137/S0895480102417768

1. Introduction.

Multilevel distance labeling can be regarded as an extension of distance two labeling which is motivated by the channel assignment problem introduced by Hale [10]. For a set of given cities (or stations), the task is to assign to each city a channel, which is a nonnegative integer, so that interference is prohibited, and the span of the channels assigned is minimized.

Usually, the level of interference between any two stations is closely related to the geographic locations of the stations—the closer the stations are, the stronger the interference is. Suppose we consider two levels of interference, major and minor. Major interference occurs between two *very close* stations; to avoid it, the channels assigned to a pair of very close stations have to be at least two apart. Minor interference occurs between *close* stations; to avoid it, the channels assigned to close stations have to be different.

To model this problem, we construct a graph G by representing each station by a vertex and connecting two vertices by an edge if the geographical locations of the corresponding stations are *very close*. Two close stations are represented by, in the corresponding graph G, a pair of vertices that are distance two apart.

Let $d_G(u, v)$ denote the distance (the shortest length of a path) between u and v in G (or simply $d(u, v)$ when G is clear in the context). Thus, for a graph G, a distance two labeling (or $L(2, 1)$-labeling) with span k is a function, $f : V(G) \to \{0, 1, 2, \ldots, k\}$, such that the following are satisfied: (1) $|f(x)-f(y)| \geq 2$ if $d(x, y) = 1$ and (2) $|f(x) - f(y)| \geq 1$ if $d(x, y) = 2$.

Distance two labeling has been studied extensively in the past decade (cf. [1, 2, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15]). One of the main research focuses has been the λ-number for a graph G, denoted by $\lambda(G)$, which is the smallest span k of a distance two labeling for G.

*Received by the editors November 13, 2002; accepted for publication (in revised form) February 26, 2005; published electronically November 15, 2005.

http://www.siam.org/journals/sidma/19-3/41776.html

†Department of Mathematics, California State University, Los Angeles, Los Angeles, CA 90032 (dliu@calstatela.edu). The work of this author was supported in part by the NSF under grants DMS 9805945 and DMS 0302456.

‡Department of Applied Mathematics, National Sun Yat-sen University, Kaohsiung, Taiwan 80424 (zhu@math.nsysu.edu.tw). The work of this author was supported in part by the National Science Council, Taiwan, under grant NSC 91-2115-M-110-003.

Practically, interference among channels might go beyond two levels. We consider interference levels from one through the largest possible value—the *diameter* of G, denoted by $\mathrm{diam}(G)$, which is the largest distance between two vertices of G.

A *multilevel distance labeling* (or *distance labeling* for short), with span k, is a function $f : V(G) \to \{0, 1, 2, \ldots, k\}$, so that for any vertices u and v,

$$|f(u) - f(v)| \geq \mathrm{diam}(G) - d_G(u, v) + 1.$$

The *radio number* (as suggested by the FM radio frequency assignment [4]) for G, denoted by $rn(G)$, is the minimum span of a distance labeling for G. Note that if $\mathrm{diam}(G) = 2$, then distance two labeling coincides with multilevel distance labeling, and in this case, $\lambda(G) = rn(G)$.

Besides its motivation from the channel assignment problem, distance labeling itself is an interesting, relatively new notion in graph coloring and worthy of further investigation for its own sake. It is surprising that determining the radio number seems a difficult problem even for some basic families of graphs. For instance, the radio number for paths and cycles has been studied by Chartrand et al. [3] and Chartrand, Erwin, and Zhang [4]. In [4] and [3], some bounds of the radio numbers for paths and cycles, respectively, were presented, while the exact values remained unknown at that time.

In this article, we completely determine the radio numbers for paths and cycles. Note that, to be consistent with distance two labelings, we allow 0 to be used as a color (or channel). However, in [4, 3] only positive integers can be used as colors. Therefore, the radio number defined in this article is *one less* than the radio number defined in [4, 3]. Being consistent throughout the article, we make necessary adjustments, reflecting this "one less" difference, for all the results quoted from [4, 3].

2. The radio number for paths.

Let P_n be the path on n vertices. Chartrand, Erwin, and Zhang [4] proved the following upper bounds for $rn(P_n)$.

THEOREM 1 (see [4]). *For any positive integer n,*

$$rn(P_n) \leq \begin{cases} 2k^2 + k & \text{if } n = 2k + 1, \\ 2(k^2 - k) + 1 & \text{if } n = 2k. \end{cases}$$

Moreover, the bound is sharp when $n \leq 5$.

In this section, we completely determine the radio numbers for paths. We first prove the following lemma.

LEMMA 2. *Let P_n be a path with vertex set $V(P_n) = \{v_1, v_2, \ldots, v_n\}$, in which $v_i \sim v_{i+1}$ for $i = 1, 2, \ldots, n-1$. Let f be an assignment of distinct nonnegative integers to $V(P_n)$. Let $(x_1, x_2, \ldots, x_n)$ be the ordering of $V(P_n)$ such that $f(x_i) < f(x_{i+1})$. The following three statements are equivalent.*

(1) *For any $1 \leq i \leq n - 2$, $\min\{d(x_i, x_{i+1}), d(x_{i+1}, x_{i+2})\} \leq n/2$.*

(2) *If $f(x_{i+1}) - f(x_i) \geq n - d(x_i, x_{i+1})$ for all $1 \leq i \leq n - 1$, then f is a distance labeling.*

(3) *If $f(x_{i+1}) - f(x_i) = n - d(x_i, x_{i+1})$ for all $1 \leq i \leq n - 1$, then f is a distance labeling.*

Proof. Note that $\mathrm{diam}(P_n) = n - 1$.

$\boxed{(1) \Rightarrow (2)}$ Assume (1) For any $1 \leq i \leq n - 2$, $\min\{d(x_i, x_{i+1}), d(x_{i+1}, x_{i+2})\} \leq n/2$ and (2) $f(x_{i+1}) - f(x_i) \geq n - d(x_i, x_{i+1})$ for all $1 \leq i \leq n - 1$. We need to show that for any $i \neq j$, $|f(x_i) - f(x_j)| \geq n - d(x_i, x_j)$.

For each $i = 1, 2, \ldots, n-1$, set

$$f_i = f(x_{i+1}) - f(x_i).$$

Assume $i < j$. Then

$$f(x_j) - f(x_i) = f_i + f_{i+1} + \cdots + f_{j-1}.$$

Assumptions (1) and (2) imply that $f_i \geq n - d(x_i, x_{i+1})$, $f_{i+1} \geq n - d(x_{i+1}, x_{i+2})$, and for any i,

$$\max\{f_i, f_{i+1}\} \geq n/2.$$

Thus, if $j \geq i+4$, then $f(x_j) - f(x_i) \geq n > n - d(x_i, x_j)$, and we are done. It suffices to consider the cases that $j = i+2$ or $j = i+3$.

Assume $j = i+2$. Without loss of generality, we may assume that $d(x_i, x_{i+1}) \geq d(x_{i+1}, x_{i+2})$, and hence $d(x_{i+1}, x_{i+2}) \leq n/2$. Since $d(x_i, x_{i+2}) \geq d(x_i, x_{i+1}) - d(x_{i+1}, x_{i+2})$, we have

$$\begin{aligned} f(x_j) - f(x_i) &= f_i + f_{i+1} \\ &\geq (n - d(x_i, x_{i+1})) + (n - d(x_{i+1}, x_{i+2})) \\ &= 2n - 2d(x_{i+1}, x_{i+2}) - (d(x_i, x_{i+1}) - d(x_{i+1}, x_{i+2})) \\ &\geq n - d(x_i, x_{i+2}). \end{aligned}$$

Assume $j = i+3$. If the sum of some pair of the distances $d(x_i, x_{i+1})$, $d(x_{i+1}, x_{i+2})$, and $d(x_{i+2}, x_{i+3})$ is at most n, then $f(x_{i+3}) - f(x_i) = f_i + f_{i+1} + f_{i+2} \geq n$, so we are done.

Thus, we assume that the sum of every pair of the distances $d(x_i, x_{i+1})$, $d(x_{i+1}, x_{i+2})$, and $d(x_{i+2}, x_{i+3})$ is greater than n. This implies that

$$d(x_{i+1}, x_{i+2}) \leq n/2 \quad \text{and} \quad d(x_i, x_{i+1}), d(x_{i+2}, x_{i+3}) > n/2.$$

Let $x_i = v_a$, $x_{i+1} = v_b$, $x_{i+2} = v_c$, $x_{i+3} = v_d$. Let m and m' be, respectively, the maximum and the minimum of $\{a, b, c, d\}$. Then $\{m, m'\} = \{a, d\}$. For otherwise, say $m' = b$, then we have $b < c < d$, implying that $d(x_{i+1}, x_{i+2}) + d(x_{i+2}, x_{i+3}) \leq n$, which is contrary to our assumption. Hence, one has

$$d(x_i, x_{i+3}) = d(x_i, x_{i+1}) + d(x_{i+2}, x_{i+3}) - d(x_{i+1}, x_{i+2}) > n/2.$$

So, $f(x_{i+3}) - f(x_i) = f_i + f_{i+1} + f_{i+2} > f_{i+1} \geq n/2 > n - d(x_i, x_{i+3})$.

$\boxed{(2) \Rightarrow (3)}$ Trivial.

$\boxed{(3) \Rightarrow (1)}$ Let $f(x_1) = 0$, and let $f(x_i) = f(x_{i-1}) + n - d(x_i, x_{i+1})$ for all i. By (3), f is a distance labeling of P_n. Assume, to the contrary of (1), that there is an index i such that

$$\min\{d(x_i, x_{i+1}), d(x_{i+1}, x_{i+2})\} > n/2.$$

Without loss of generality, we assume that $d(x_i, x_{i+1}) \geq d(x_{i+1}, x_{i+2})$. Then

$$d(x_i, x_{i+2}) = d(x_i, x_{i+1}) - d(x_{i+1}, x_{i+2}),$$

and thus

$$\begin{aligned} f(x_{i+2}) - f(x_i) &= n - d(x_i, x_{i+1}) + n - d(x_{i+1}, x_{i+2}) \\ &= 2n - 2(d(x_{i+1}, x_{i+2})) - d(x_i, x_{i+2}) \\ &< n - d(x_i, x_{i+2}), \end{aligned}$$

contrary to the assumption that f is a distance labeling. □

THEOREM 3. *For any $n \geq 3$,*

$$rn(P_n) = \begin{cases} 2k^2 + 2 & \text{if } n = 2k+1, \\ 2k(k-1) + 1 & \text{if } n = 2k. \end{cases}$$

Proof. Note that, for even paths, by Theorem 1 it suffices to show that $rn(P_{2k}) \geq 2k(k-1) + 1$. However, for completeness, we present a proof here without using Theorem 1.

First, we show that $rn(P_{2k+1}) \leq 2k^2 + 2$ and $rn(P_{2k}) \leq 2k(k-1) + 1$. Assume $P_{2k+1} = (v_1, v_2, \ldots, v_{2k+1})$, where $v_i \sim v_{i+1}$. Order the vertices of P_{2k+1} as follows:

$$v_k, v_{k+k}, v_1, v_{1+k}, v_{1+k+k}, v_3, v_{3+k}, v_4, v_{4+k}, v_5, v_{5+k}, \ldots, v_{k-1}, v_{k-1+k}, v_2, v_{2+k}.$$

Rename the vertices of P in the above ordering by $x_1, x_2, \ldots, x_{2k+1}$. Namely, let $x_1 = v_k$, $x_2 = v_{k+k}, \ldots, x_{2k+1} = v_{2+k}$.

Let f be the mapping defined as $f(x_1) = 0$, and for $i = 2, 3, \ldots, 2k+1$,

$$f(x_i) = f(x_{i-1}) + 2k + 1 - d(x_{i-1}, x_i).$$

It is easy to verify that the ordering and the mapping f satisfy the conditions of Lemma 2(1) and (3). Therefore f is a distance labeling of P_{2k+1}.

It remains to show that $f(x_{2k+1}) = 2k^2 + 2$. By definition,

$$\begin{aligned} f(x_{2k+1}) &= \sum_{i=1}^{2k} [2k + 1 - d(x_i, x_{i+1})] \\ &= 2k(2k+1) - \sum_{i=1}^{2k} d(x_i, x_{i+1}). \end{aligned}$$

Thus, it suffices to show that

$$\sum_{i=1}^{2k} d(x_i, x_{i+1}) = 2k^2 + 2k - 2.$$

Note that if $x_i = v_j$ and $x_{i+1} = v_{j'}$, then $d(x_i, x_{i+1}) = |j - j'|$, which is equal to either $j - j'$ or $j' - j$, whichever is positive. By replacing each term $d(x_i, x_{i+1})$ with the corresponding $j - j'$ or $j' - j$, whichever is positive, we obtain a summation whose entries are $\pm j$ for $j \in \{1, 2, \ldots, 2k+1\}$.

For the ordering above, if $j \leq k$, then the vertex preceding v_j is $v_{j'}$ for some $j' \geq k+2$, and the vertex following v_j is $v_{j''}$ for some $j'' \geq k+1$. Therefore, for each $1 \leq j \leq k$, whenever $\pm j$ occurs in the summation above, it occurs as a $-j$. Similarly, if $k+2 \leq j \leq 2k+1$, then whenever $\pm j$ occurs in the summation it occurs as a $+j$. The number $k+1$ occurs once as $+(k+1)$ and once as $-(k+1)$. Also it is easy to see that each j occurs twice in the summation, except that each of $j = k$ and $j = k + 2$ occurs only once in the summation. Hence, we have

$$\sum_{i=1}^{2k} d(x_i, x_{i+1}) = 2\left(\sum_{j=k+2}^{2k+1} j - \sum_{j=1}^{k} j\right) - (k + 2 - k)$$

$$= 2k^2 + 2k - 2.$$

The case for even paths is similar. Order the vertices of P_{2k} as follows:

$$v_k, v_{k+k}, v_2, v_{2+k}, v_3, v_{3+k}, \ldots, v_{k-1}, v_{k-1+k}, v_1, v_{1+k}.$$

Rename the vertices so that the ordering above is $x_1, x_2, \ldots, x_{2k}$. Namely, let $x_1 = v_k, x_2 = v_{k+k}, \ldots, x_{2k} = v_{1+k}$.

Let f be the mapping defined as $f(x_1) = 0$, and for $i = 2, 3, \ldots, 2k$,

$$f(x_i) = f(x_{i-1}) + 2k - d(x_{i-1}, x_i).$$

Then the ordering and the mapping f satisfy the conditions of Lemma 2(1) and (3). Therefore f is a distance labeling of P_{2k}.

Similarly, in the summation $\sum_{i=1}^{2k-1} d(x_i, x_{i+1})$, each $j \in \{1, 2, \ldots, k - 1\}$ occurs twice as $-j$, k occurs once as a $-k$, each of $j \in \{k + 2, k + 3, \ldots, 2k\}$ occurs twice as $+j$, and $k + 1$ occurs once as a $+(k + 1)$. Therefore,

$$\sum_{i=1}^{2k-1} d(x_i, x_{i+1}) = 2\left(\sum_{j=k+2}^{2k} j - \sum_{j=1}^{k-1} j\right) + k + 1 - k$$

$$= 2k^2 - 1.$$

This implies

$$f(x_{2k}) = \sum_{i=1}^{2k-1} [2k - d(x_i, x_{i+1})]$$

$$= 2k(2k - 1) - \sum_{i=1}^{2k-1} d(x_i, x_{i+1})$$

$$= 4k^2 - 2k - 2k^2 + 1$$
$$= 2k(k - 1) + 1.$$

Next, we show that $rn(P_{2k+1}) \geq 2k^2 + 2$. Let f be a distance labeling of P_{2k+1}. Order the vertices of P_{2k+1} as $x_1, x_2, \ldots, x_{2k+1}$ such that $f(x_i) < f(x_{i+1})$ for all i. Assume $x_i = v_{\sigma(i)}$. Then σ is a permutation of $\{1, 2, \ldots, 2k + 1\}$. We shall prove that $f(x_{2k+1}) \geq 2k^2 + 2$.

By definition, $f(x_1) \geq 0$ and $f(x_i) \geq f(x_{i-1}) + 2k + 1 - d(x_{i-1}, x_i)$ for $i = 2, 3, \ldots, 2k + 1$. Thus

$$f(x_{2k+1}) \geq \sum_{i=1}^{2k} [2k + 1 - d(x_i, x_{i+1})]$$

$$= 2k(2k + 1) - \sum_{i=1}^{2k} d(x_i, x_{i+1}).$$

If $\sum_{i=1}^{2k} d(x_i, x_{i+1}) \le 2k^2 + 2k - 2$, then $f(x_{2k+1}) \ge 2k^2 + 2$, and we are done. Hence, assume $\sum_{i=1}^{2k} d(x_i, x_{i+1}) > 2k^2 + 2k - 2$.

Claim 1. If $\sum_{i=1}^{2k} d(x_i, x_{i+1}) > 2k^2 + 2k - 2$, then $\sum_{i=1}^{2k} d(x_i, x_{i+1}) = 2k^2 + 2k - 1$ and there is an index i such that $f(x_{i+1}) - f(x_i) \ge n - d(x_{i+1}, x_i) + 1$.

Proof of Claim 1. Note that $d(x_i, x_{i+1})$ is equal to either $\sigma(i) - \sigma(i+1)$ or $\sigma(i+1) - \sigma(i)$, whichever is positive. By replacing each term $d(x_i, x_{i+1})$ with the corresponding $\sigma(i) - \sigma(i+1)$ or $\sigma(i+1) - \sigma(i)$, whichever is positive, we obtain a summation whose entries are $\pm j$ for $j \in \{1, 2, \dots, 2k+1\}$.

All together, there are $4k$ terms in the summation $\sum_{i=1}^{2k} d(x_i, x_{i+1})$, half of them positive and half negative. Each $j \in \{1, 2, \dots, 2k+1\}$ occurs as $\pm j$ exactly twice in the summation, except for two values each of which occurs only once.

To maximize the summation $\sum_{i=1}^{2k} d(x_i, x_{i+1})$, one needs to minimize the absolute values for the negative terms while maximizing the values of the positive terms. It is easy to verify that there are two combinations achieving the maximum summation:

Case 1. Each of the numbers in $\{k+2, k+3, k+4, \dots, 2k+1\}$ occurs twice as a positive, each of $\{1, 2, \dots, k-1\}$ occurs twice as a negative, and each of k and $k+1$ occurs once as a negative.

Case 2. Each of the numbers in $\{k+3, k+4, \dots, 2k+1\}$ occurs twice as a positive, each of $\{1, 2, \dots, k\}$ occurs twice as a negative, and each of $k+1$ and $k+2$ occurs once as a positive.

In both cases, we have

$$\sum_{i=1}^{2k} d(x_i, x_{i+1}) = 2k^2 + 2k - 1.$$

In Case 1, we must have $\{\sigma(1), \sigma(2k+1)\} = \{k+1, k\}$. Moreover, $\sigma(i) \ge k+2$ if and only if $\sigma(i+1) \le k+1$. In particular, if $\sigma(i) = 1$, then $\sigma(i-1) \ge k+2$ and $\sigma(i+1) \ge k+2$. This violates (1) in Lemma 2. As f is a distance labeling, it follows from Lemma 2(3) that there exists some i such that $f(x_{i+1}) - f(x_i) \ge n - d(x_i, x_{i+1}) + 1$.

In Case 2, we must have $\{\sigma(1), \sigma(2k+1)\} = \{k+1, k+2\}$. Moreover, $\sigma(i) \ge k+1$ if and only if $\sigma(i+1) \le k$. In particular, if $\sigma(i) = 2k+1$, then $\sigma(i-1) \le k$ and $\sigma(i+1) \le k$. Again, this violates (1) in Lemma 2, and it follows from Lemma 2(3) that there exists some i such that $f(x_{i+1}) - f(x_i) \ge n - d(x_i, x_{i+1}) + 1$. □

By some calculation, it follows from Claim 1 that if $\sum_{i=1}^{2k} d(x_i, x_{i+1}) > 2k^2 + 2k - 2$, we also have $f(x_{2k+1}) \ge 2k^2 + 2$, completing the proof for odd paths.

We now show that $rn(P_{2k}) \ge 2(k^2 - k) + 1$. Let f be a distance labeling of P_{2k}. Let $x_1, x_2, \dots, x_{2k}$ be the ordering of the vertices of P_{2k} such that $f(x_i) < f(x_{i+1})$ for all i. Then

$$\begin{aligned} f(x_{2k}) &\ge \sum_{i=1}^{2k-1} [2k - d(x_i, x_{i+1})] \\ &= 2k(2k-1) - \sum_{i=1}^{2k-1} d(x_i, x_{i+1}). \end{aligned}$$

Similarly, in the summation $\sum_{i=1}^{2k-1} d(x_i, x_{i+1})$, each $j \in \{1, 2, \dots, 2k\}$ occurs twice as $\pm j$, except for two values which each occur only once. Moreover, $2k-1$ of the terms are positive and $2k-1$ of them are negative. Thus to maximize the summation subject to the constraint, each number in $\{1, 2, \dots, k-1\}$ occurs twice as negative terms, and each number in $\{k+2, k+3, \dots, 2k\}$ occurs twice as positive terms, while k and $k+1$ occur once, respectively, as a negative term and a positive term. Hence, we have

$$\sum_{i=1}^{2k-1} d(x_i, x_{i+1}) \leq 2(k^2-1)+1,$$

implying

$$f(x_{2k}) \geq 2k(2k-1) - 2(k^2-1) - 1 \geq 2k(k-1)+1. \qquad \square$$

3. The radio number for cycles. Let C_n denote the cycle on n vertices. Chartrand et al. [3] proved the following bounds for $rn(C_n)$.

THEOREM 4 (see [3]). *For $k \geq 3$,*

$$rn(C_n) \leq \begin{cases} k^2 & \textit{if } n = 2k+1, \\ k^2-k+1 & \textit{if } n = 2k. \end{cases}$$

Moreover, $rn(C_n) \geq 3\lceil \frac{n}{2} - 1 \rceil - 1$ for $n \geq 6$.

In this section, we completely determine the radio number for cycles. For any integer $n \geq 3$, let

$$\phi(n) = \begin{cases} k+1 & \text{if } n = 4k+1, \\ k+2 & \text{if } n = 4k+r \text{ for some } r = 0, 2, 3. \end{cases}$$

THEOREM 5. *Let C_n be the n-vertex cycle, $n \geq 3$. Then*

$$rn(C_n) = \begin{cases} \frac{n-2}{2}\phi(n)+1 & \textit{if } n \equiv 0, 2 \pmod 4, \\ \frac{n-1}{2}\phi(n) & \textit{if } n \equiv 1, 3 \pmod 4. \end{cases}$$

First we prove that the desired numbers in Theorem 5 are lower bounds for $rn(C_n)$. Assume $V(C_n) = \{v_0, v_1, v_2, \dots, v_{n-1}\}$, where $v_i \sim v_{i+1}$ and $v_{n-1} \sim v_0$. Let f be a distance labeling for C_n. We order the vertices of $V(C_n)$ by $x_0, x_1, x_2, \dots, x_{n-1}$ with $f(x_i) < f(x_{i+1})$.

Denote $d = \operatorname{diam}(C_n)$. Then $d = \lfloor n/2 \rfloor$. For $i = 0, 1, 2, \dots, n-2$, set

$$d_i = d(x_i, x_{i+1}) \text{ and } f_i = f(x_{i+1}) - f(x_i).$$

By definition, $f_i \geq d - d_i + 1$ for all i.

To proceed with the proof of Theorem 5, we need the following two results.

LEMMA 6. *For any $0 \leq i \leq n-3$, $f_i + f_{i+1} \geq \phi(n)$.*

Proof. Assume to the contrary that for some i, $f_i + f_{i+1} \leq \phi(n) - 1$. Then $f_i, f_{i+1} \leq \phi(n) - 2$. So, we have $d_i \geq d - f_i + 1 \geq d - \phi(n) + 3$ and $d_{i+1} \geq d - \phi(n) + 3$, implying that $d_i, d_{i+1} > d/2$. Therefore, $d(x_i, x_{i+2})$ is equal to either $|d_i - d_{i+1}|$ or $n - (d_i + d_{i+1})$. In the former case, $d(x_i, x_{i+2}) \leq d - (d - \phi(n) + 3) = \phi(n) - 3$, implying that

$$f_i + f_{i+1} = f(x_{i+2}) - f(x_i) \geq d - (\phi(n) - 3) + 1 \geq \phi(n),$$

contrary to our assumption.

If it is the latter case, then by definition all of the following hold:

$$\begin{aligned} f(x_{i+1}) - f(x_i) &\geq d - d_i + 1, \\ f(x_{i+2}) - f(x_{i+1}) &\geq d - d_{i+1} + 1, \\ f(x_{i+2}) - f(x_i) &\geq d - (n - d_i - d_{i+1}) + 1. \end{aligned}$$

Hence, $2(f(x_{i+2}) - f(x_i)) \geq 3d - n + 3$. Easy calculation shows that $f_i + f_{i+1} = f(x_{i+2}) - f(x_i) \geq \phi(n)$, a contradiction. □

Corollary 7. *For any integer $n \geq 3$,*

$$rn(C_n) \geq \begin{cases} \frac{n-2}{2}\phi(n) + 1 & \textit{if } n \equiv 0, 2 \pmod 4, \\ \frac{n-1}{2}\phi(n) & \textit{if } n \equiv 1, 3 \pmod 4. \end{cases}$$

Proof. If $n = 4k$ or $n = 4k + 2$, by Lemma 6, the span of a distance labeling f for C_n is

$$f(x_{n-1}) = \sum_{i=0}^{n-2} f_i = \sum_{i=0}^{(n-4)/2} (f_{2i} + f_{2i+1}) + f_{n-2} \geq \frac{n-2}{2}\phi(n) + 1.$$

If $n = 4k + 1$ or $n = 4k + 3$, by Lemma 6 the span of a distance labeling f for C_n is

$$f(x_{n-1}) = \sum_{i=0}^{n-2} f_i = \sum_{i=0}^{(n-3)/2} (f_{2i} + f_{2i+1}) \geq \frac{n-1}{2}\phi(n). \qquad \square$$

To complete the proof of Theorem 5, it remains to find distance labelings for C_n with spans equal to the desired numbers. We consider four cases. For each case, we present a distance labeling f of C_n, achieving the bound.

In each of the four cases, the labeling is generated by two sequences, the *distance gap sequence*

$$D = (d_0, d_1, d_2, d_3, \dots, d_{n-2})$$

and the *color gap sequence*

$$F = (f_0, f_1, f_2, \dots f_{n-2}).$$

The distance gap sequence, in which each $d_i \leq d$ is a positive integer, is used to generate an ordering of the vertices of C_n. Let $\tau : \{0, 1, \dots, n-1\} \to \{0, 1, \dots, n-1\}$ be defined as $\tau(0) = 0$ and

$$\tau(i + 1) = \tau(i) + d_i \pmod n.$$

We will show that for each of the distance sequences given below, the corresponding τ is a permutation. Let $x_i = v_{\tau(i)}$ for $i = 0, 1, 2, \dots, n-1$. Then $x_0, x_1, \dots, x_{n-1}$ is an ordering of the vertices of C_n. Since $1 \leq d_i \leq d$ for each i, we have $d(x_i, x_{i+1}) = d_i$.

The color gap sequence is used to assign labels to the vertices of C_n. Let f be the labeling defined by $f(x_0) = 0$, and for $i \geq 1$,

$$f(x_{i+1}) = f(x_i) + f_i.$$

Since $f_i = f(x_{i+1}) - f(x_i)$ and $d(x_i, x_{i+1}) = d_i$, to show that f is indeed a distance labeling, it suffices to prove that all of the following hold for any i:

(1) τ is a permutation,
(2) $f_i \geq d - d_i + 1$,
(3) $f_i + f_{i+1} \geq d - d(x_i, x_{i+2}) + 1$,
(4) $f_i + f_{i+1} + f_{i+2} \geq d - d(x_i, x_{i+3}) + 1$,
(5) $f_i + f_{i+1} + f_{i+2} + f_{i+3} \geq d$.

For all the labelings given below, (5) is trivial, (2) is obvious, and (3) and (4) are also easy to verify. In all the cases, we sketch a proof for (1), and leave it to the reader to verify (2)–(5).

$\boxed{\text{Case 1. } n = 4k}$ In this case, $d = 2k$. The distance gap sequence D is given by

$$d_i = \begin{cases} 2k & \text{if } i \text{ is even,} \\ k & \text{if } i \equiv 1 \pmod 4, \\ k+1 & \text{if } i \equiv 3 \pmod 4. \end{cases}$$

The color gap sequence F is given by

$$f_i = \begin{cases} 1 & \text{if } i \text{ is even,} \\ k+1 & \text{if } i \text{ is odd.} \end{cases}$$

Then we have, for $i = 0, 1, 2, \ldots, k-1$,

$$\begin{array}{lcll} \tau(4i) & = & 2ik + i & \pmod n, \\ \tau(4i+1) & = & (2i+2)k + i & \pmod n, \\ \tau(4i+2) & = & (2i+3)k + i & \pmod n, \\ \tau(4i+3) & = & (2i+1)k + i & \pmod n. \end{array}$$

We prove that τ is a permutation. Assume to the contrary that $\tau(4i + j) = \tau(4i'+j')$ for some $i, i' \in \{0, 1, 2, \ldots, k-1\}$ and $j, j' \in \{0, 1, 2, 3\}$ with $4i+j < 4i'+j'$. Then, clearly $i < i'$ and

$$(2i + t)k + i \equiv (2i' + t')k + i' \pmod n \text{ for some } t, t' = 0, 1, 2, 3.$$

Therefore, we have $2(i' - i)k + (t' - t)k \equiv i - i' \pmod n$, which is impossible, as $0 < i' - i < k$ and $2(i' - i)k + (t' - t)k \equiv sk \pmod n$ for some integer s.

The span of f is equal to $f_0 + f_1 + f_2 + \cdots + f_{n-2} = (k + 2)(2k - 1) + 1$.

$\boxed{\text{Case 2. } n = 4k + 2}$ In this case, $d = 2k+1$. The distance gap sequence D is defined by

$$d_i = \begin{cases} 2k+1 & \text{if } i \text{ is even,} \\ k+1 & \text{if } i \text{ is odd.} \end{cases}$$

The color gap sequence F is defined by

$$f_i = \begin{cases} 1 & \text{if } i \text{ is even,} \\ k+1 & \text{if } i \text{ is odd.} \end{cases}$$

Hence, for $i = 0, 1, \ldots, 2k$, we have

$$\begin{array}{lcll} \tau(2i) & = & i(3k+2) & \pmod n, \\ \tau(2i+1) & = & i(3k+2) + 2k + 1 & \pmod n. \end{array}$$

We show that τ is a permutation. Note that $(n, k) \leq 2$ and $3k+2 \equiv -k \pmod n$. Thus, $(i - i')(3k + 2) \equiv k(i' - i) \not\equiv 0 \pmod n$ if $0 < i - i' < n/2$. This implies that $\tau(2i) \neq \tau(2i')$ and $\tau(2i + 1) \neq \tau(2i' + 1)$ if $i \neq i'$.

If $\tau(2i) = \tau(2i'+1)$, then similarly, we get $(i-i')k \equiv 2k+1 = n/2 \pmod{n}$. Since $\gcd(n/2, k) = 1$ and $|i-i'| \leq 2k < n/2$, this is impossible.

The span of f is $f_0 + f_1 + \cdots + f_{n-2} = 2k(k+2)+1$.

$\boxed{\text{Case 3. } n = 4k+1}$ In this case, $d = 2k$. The distance gap sequence D is defined by

$$d_{4i} = d_{4i+2} = 2k - i \text{ and } d_{4i+1} = d_{4i+3} = k+1+i.$$

The color gap sequence F is defined by

$$f_i = d - d_i + 1 = 2k - d_i + 1.$$

Then, the mapping τ on the vertices of C_n has

$$\begin{aligned}
\tau(2i) &= i(3k+1) \pmod{n} \\
&= -ik \pmod{n}, && 0 \leq i \leq 2k, \\
\tau(4i+1) &= 2i(3k+1) + 2k - i \pmod{n} \\
&= 2(i+1)k \pmod{n}, && 0 \leq i \leq k-1, \\
\tau(4i+3) &= (2i+1)(3k+1) + 2k - i \pmod{n} \\
&= (2i+1)k \pmod{n}, && 0 \leq i \leq k-1.
\end{aligned}$$

We show that τ is indeed a permutation. Let

$$\begin{aligned}
S = &\{-i : 0 \leq i \leq 2k\} \cup \{2(i+1) : 0 \leq i \leq k-1\} \\
&\cup \{2i+1 : 0 \leq i \leq k-1\} \\
= &\{-2k, -(2k-1), \ldots, 0, 1, \ldots, 2k\}.
\end{aligned}$$

By the definition of τ, for any $0 \leq j \leq 4k$, we have $\tau(j) = a_j k \pmod{n}$ for some $a_j \in S$ and $a_j \neq a_{j'}$ if $j \neq j'$. Thus to prove $\tau(j) \neq \tau(j')$ for $j \neq j'$, it suffices to show that for any distinct elements a, a' of S, $ak \neq a'k \pmod{n}$. This is obvious, as $(n,k) = 1 \pmod{n}$ and for any two distinct elements a, a' of S, $0 < |a - a'| < n$. So $(a-a')k \not\equiv 0 \pmod{n}$, and hence τ is a permutation.

Using the fact that $d_{2i} + d_{2i+1} = 3k+1$ for any i, the span of f is

$$\begin{aligned}
f_0 + f_1 + f_2 + \cdots + f_{n-2} &= (4k)(2k) - (d_0 + d_1 + \cdots + d_{n-2}) + 4k \\
&= 8k^2 - 2k(3k+1) + 4k \\
&= 2k(k+1).
\end{aligned}$$

$\boxed{\text{Case 4. } n = 4k+3}$ In this case, $d = 2k+1$. The distance gap sequence D is defined by

$$d_{4i} = d_{4i+2} = 2k+1-i, \quad d_{4i+1} = k+1+i, \quad d_{4i+3} = k+2+i.$$

The coloring gap sequence F is

$$f_i = \begin{cases} d - d_i + 1 = 2k - d_i + 2, & i \not\equiv 3 \pmod{4}, \\ d - d_i + 2 = 2k - d_i + 3 & \text{otherwise.} \end{cases}$$

Then the mapping τ on the vertices of C_n has

$$\begin{array}{lcll}
\tau(4i) & = & i(6k+5) \pmod n & \\
& = & 2i(k+1) \pmod n, & 0 \le i \le k, \\[1em]
\tau(4i+1) & = & 2i(k+1)+2k+1-i \pmod n & \\
& = & (i+1)(2k+1) \pmod n & \\
& = & -2(i+1)(k+1) \pmod n, & 0 \le i \le k, \\[1em]
\tau(4i+2) & = & (i+1)(2k+1)+k+1+i \pmod n & \\
& = & (i+1)(2k+2)+k \pmod n & \\
& = & 2(i+1)(k+1)-3(k+1) \pmod n & \\
& = & (2i-1)(k+1) \pmod n, & 0 \le i \le k, \\[1em]
\tau(4i+3) & = & 2i(k+1)+3k+2+2k+1-i \pmod n & \\
& = & i(2k+1)+k \pmod n & \\
& = & -i(2k+2)-3(k+1) \pmod n & \\
& = & -(2i+3)(k+1) \pmod n, & 0 \le i \le k-1.
\end{array}$$

Now we prove that τ is a permutation. Let

$$\begin{aligned}
S &= \{2i : 0 \le i \le k\} \cup \{-2(i+1) : 0 \le i \le k\} \\
&\quad \cup \{2i-1 : 0 \le i \le k\} \cup \{-(2i+3) : 0 \le i \le k-1\} \\
&= \{-(2k+2), -(2k+1), \ldots, 0, 1, \ldots, 2k\}.
\end{aligned}$$

By the definition of τ, for any $0 \le j \le 4k+2$, we have $\tau(j) = a_j(k+1) \pmod n$ for some $a_j \in S$, and $a_j \ne a_{j'}$ if $j \ne j'$. Thus, to prove $\tau(j) \ne \tau(j')$ for $j \ne j'$, it suffices to show that for any distinct elements a, a' of S, $a(k+1) \ne a'(k+1) \pmod n$. This is obvious, as $(n, k+1) = 1 \pmod n$ and for any two distinct elements a, a' of S, $0 < |a - a'| < n$. Hence, τ is a permutation.

The span of f is

$$\begin{aligned}
f_0 + f_1 + \cdots + f_{n-2} &= 2k(4k+2) - (d_0 + d_1 + \cdots + d_{n-2}) + 2(4k+2) + k \\
&= 2k(4k+2) - [k(6k+5) + 3k + 2] + 9k + 4 \\
&= (k+2)(2k+1).
\end{aligned}$$

This completes the proof of Theorem 5. □

Acknowledgment. The authors wish to thank the referees for their careful reading and constructive comments on earlier versions of this article, which resulted in better presentation of this article.

REFERENCES

[1] G. Chang, C. Ke, D. Kuo, D. Liu, and R. Yeh, *A generalized distance two labeling of graphs*, Discrete Math., 220 (2000), pp. 57–66.

[2] G. J. Chang and D. Kuo, *The $L(2,1)$-labeling problem on graphs*, SIAM J. Discrete Math., 9 (1996), pp. 309–316.

[3] G. Chartrand, D. Erwin, F. Harary, and P. Zhang, *Radio labelings of graphs*, Bull. Inst. Combin. Appl., 33 (2001), pp. 77–85.

[4] G. Chartrand, D. Erwin, and P. Zhang, *A graph labeling problem suggested by FM channel restrictions*, Bull. Inst. Combin. Appl., 43 (2005), pp. 43–57.

[5] J. Georges and D. Mauro, *Generalized vertex labelings with a condition at distance two*, Congr. Numer., 109 (1995), pp. 141–159.

[6] J. P. Georges, D. W. Mauro, and M. I. Stein, *Labeling products of complete graphs with a condition at distance two*, SIAM J. Discrete Math., 14 (2001), pp. 28–35.

[7] J. Georges, D. Mauro, and M. Whittlesey, *On the size of graphs labeled with a condition at distance two*, J. Graph Theory, 22 (1996), pp. 47–57.

[8] J. Georges, D. Mauro, and M. Whittlesey, *Relating path covering to vertex labelings with a condition at distance two*, Discrete Math., 135 (1994), pp. 103–111.

[9] J. R. Griggs and R. K. Yeh, *Labelling graphs with a condition at distance* 2, SIAM J. Discrete Math., 5 (1992), pp. 586–595.

[10] W. K. Hale, *Frequency assignment: Theory and applications*, Proc. IEEE, 68 (1980), pp. 1497–1514.

[11] J. van den Heuvel, R. Leese, and M. Shepherd, *Graph labeling and radio channel assignment*, J. Graph Theory, 29 (1998), pp. 263–283.

[12] D. Liu and R. K. Yeh, *On distance two labellings of graphs*, Ars Combin., 47 (1997), pp. 13–22.

[13] D. D.-F. Liu and X. Zhu, *Circular distance two labeling and the λ-number for outerplanar graphs*, SIAM J. Discrete Math., 19 (2005), pp. 281–293.

[14] D. Sakai, *Labeling chordal graphs: Distance two condition*, SIAM J. Discrete Math., 7 (1994), pp. 133–140.

[15] M. A. Whittlesey, J. P. Georges, and D.W. Mauro, *On the λ-number of Q_n and related graphs*, SIAM J. Discrete Math., 8 (1995), pp. 499–506.

SIAM J. Discrete Math.
Vol. 19, No. 3, pp. 622–632

GRAPH-THEORETIC GENERALIZATION OF THE SECRETARY PROBLEM: THE DIRECTED PATH CASE[*]

GRZEGORZ KUBICKI[†] AND MICHAŁ MORAYNE[‡]

Abstract. We consider the following on-line decision problem. The vertices of a directed path of a known length are being observed one by one in some random order by a selector. At time t the selector examines the tth vertex and knows the directed graph induced by the t vertices that have been already examined. The selector's aim is to choose the currently examined vertex maximizing the probability that this vertex is the "uppermost" one, i.e., the only one that does not have an outgoing edge. An optimal algorithm for such a choice (in other words, optimal stopping time) is given. For a cardinality n of the directed path considered, the probability p_n of the right choice according to the optimal algorithm is given, and it is shown that $p_n\sqrt{n} \to \sqrt{\pi}/2$ as $n \to \infty$.

Key words. graph, directed path, secretary problem, best choice

AMS subject classification. 60G40

DOI. 10.1137/S0895480104440596

1. Introduction.

The celebrated secretary problem (known also as the best choice problem) can be stated as follows: there are n elements (candidates for a job of a secretary) which are linearly ordered from the worst to the best, $c_1, \ldots, c_n$, and which are observed one by one in some random permutation $c_{\pi_1}, \ldots, c_{\pi_n}$ by a selector who is to make at some moment τ an on-line decision picking the presently examined candidate c_τ maximizing the probability $P[\pi_\tau = n]$. The choice of the selector is based only on his knowledge of the relative ranks of the candidates examined so far and the number n of all candidates. At the moment of choice, the selector has no knowledge of the ranks of the future candidates.

This problem was solved in [GM]. Its potential usefulness going far beyond the entertaining original statement, and, probably, also its irresistible beauty, stimulated many authors to enrich its original content and to consider many variants of it. It seems now to be a field of research on its own, having a large bibliography and many interesting results. For a comprehensive treatment of the subject the reader is advised to consult [F] and [BG].

Further generalizations opened up when it was realized that sometimes the choice must be made from elements that are not necessarily comparable in some absolute sense. This led to a partial order version of the secretary problem that has the formulation fully analogous to that where candidates are linearly ordered. The set P to be examined is now equipped with some partial order $\prec$ and at a moment t the selector knows only the candidates $c_{\pi_1}, \ldots, c_{\pi_t}$ that have been examined so far and their relative ranks or, in other words, the induced order $\prec \cap \{c_{\pi_1}, \ldots, c_{\pi_t}\}^2$. The aim of the selector is to pick the presently considered candidate in such a way that the

[*] Received by the editors February 4, 2004; accepted for publication (in revised form) March 15, 2005; published electronically November 15, 2005.

http://www.siam.org/journals/sidma/19-3/44059.html

[†] Department of Mathematics, University of Louisville, Louisville, KY 40292 (gkubicki@louisville.edu).

[‡] Institute of Mathematics, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland (morayne@im.pwr.wroc.pl). The research of this author was partially supported by KBN grant 3 T11 C 011 26. This article was, in part, written while this author was visiting the Department of Mathematics, University of Louisville, Louisville, KY.

probability that it belongs to the set of maximal elements with respect to the partial order $\prec$ is maximal possible. In some situations giving an optimal stopping time seems to be very hard, therefore the algorithms were searched where the probability of choosing a maximal element was sufficiently high.

The partial order version of the secretary problem was, for instance, considered in the series of papers by Baryshnikov, Berezovskiy, and Gnedin, which are well surveyed in [G] (threshold stopping times), [M] (optimal stopping time for the partial orders whose Hasse diagram is a finite complete binary tree of a given length), and [P] and [KMN] (universal randomized stopping times for partially ordered sets of known cardinality in the case where the selector has no prior knowledge of the order).

All these considerations may still be viewed from a broader graph-theoretic perspective. Namely, if with a partially ordered set $(P, \prec)$ we associate its directed Hasse diagram, picking a maximal element is picking an element that is never a starting point of a directed edge. This statement, however, no longer requires a partial order structure. It is enough to deal with a directed graph. The role of maximal elements is played now by those vertices which are never starting points of directed edges. Actually, there is no obstacle to formulate a still more general problem where for a given graph (not necessarily directed) we want to choose in the online decision process described above a vertex from some predefined set of vertices. Effective algorithms for such a choice may be potentially applicable, for instance, in searches for appropriate servers which are a part of a known computer network.

In this paper we consider the case of a directed path of given cardinality n. We find an optimal algorithm (stopping time) for the choice of the maximal element.

2. Definitions and notation. For a set X let $\mathcal{P}(X)$ denote the family of all subsets of X. $\mathbf{N}$ denotes the set of positive integers. Let S_n be the family of all permutations of the set $\{1, 2, \ldots, n\}$. A *graph* is a pair (V, E), where V is a set of *vertices* and E is a family of nonempty subsets of V of cardinality at most two. Each such subset is called an *edge* (connecting its elements). In a *directed graph* (V, E) the set E is a set of ordered pairs of elements of V. Therefore every edge has a direction. A *simple graph* is a graph (directed or not) having at most one edge between any two vertices and having no edge connecting a vertex to itself. A *directed path* is a directed graph $G = (V, E)$, where $V = \{v_1, \ldots, v_n\}$, $v_i \neq v_j$ for $i \neq j$, and $E = \{(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n)\}$. A *maximal vertex* of a directed graph $G = (V, E)$ is any vertex $v \in V$ such that $(v, u) \notin E$ for every $u \in V$. The set of all maximal vertices of $G = (V, E)$ will be denoted by $\mathrm{Max}\,\mathrm{G}$ or $\mathrm{Max_E V}$ or, if E is known from the context, simply by $\mathrm{Max}\,\mathrm{V}$. Thus if (V, E) is the directed path defined above, we have $\mathrm{Max_E V} = \{v_n\}$. A graph $G = (V, E)$ (directed or not) is connected if for any $u, v \in V$ there is a sequence $u = u_0, u_1, \ldots, u_k = v$ such that there is an edge (no condition is imposed on its direction) between u_i and u_{i+1}, $0 \leq i < k$. For a graph $G = (V, E)$, its *induced subgraph* $G' = (W, E \cap W^2)$ (induced by W), $W \subset V$, is called a *connected component* if it is a maximal connected induced subgraph.

Let $G = (V, E)$ be a directed graph and $v_1, v_2, \ldots, v_k$ be a sequence of pairwise different vertices of G. Let $R \subseteq \mathbf{N}^2$. We write $(v_1, \ldots, v_k) \cong R$ if for all $i, j \leq k$, $i \neq j$, $(v_i, v_j) \in E$ if and only if $(i, j) \in R$.

Let $(\Omega, \mathcal{F}, P)$ be a probability space. Let $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \cdots \subseteq \mathcal{F}_n \subseteq \mathcal{F}$ be a sequence of σ-algebras. We call such a sequence a *filtration.* We say that a random variable $\tau : \omega \to \{1, 2, \ldots, n\}$ is a *stopping time* with respect to a filtration $(\mathcal{F}_t)_{t=1}^n$ if $\tau^{-1}(\{t\}) \in \mathcal{F}_t$ for each $t \leq n$.

If we think of $\tau(\omega)$, $\omega \in \Omega$ as a moment when to stop observing a certain process depending on ω and $t = 1, 2, \ldots, n$, then the condition $\tau^{-1}(\{t\}) \in \mathcal{F}_t$ means that our decision to stop at t is based only on the events that took place until this moment and does not depend on any information about the future events.

3. Formal model. Though in this paper we consider only the case of directed path, for the problem we want to consider it is worthwhile to give a more general probabilistic model concerning any directed graph.

Thus let $G = (V, E)$ be a fixed directed graph. Let $V = \{v_1, v_2, \ldots, v_n\}$. Let $\Omega = S_n$, where S_n is the set of all permutations of $1, 2, \ldots, n$. Let $\mathcal{F} = \mathcal{P}(\Omega)$. Let the probability measure $P : \mathcal{F} \to [0, 1]$ be defined by $P(\{\pi\}) = 1/n!$ for each $\pi \in S_n$ (i.e., we assume a uniform distribution). Let

$$\mathcal{F}_t = \sigma\{\{\pi \in \Omega : (v_{\pi_1}, v_{\pi_2}, \ldots, v_{\pi_t}) \cong R\} : R \subseteq \mathbf{N}^2\}, \quad 1 \le t \le n.$$

The extension of the secretary problem to the case of the directed graph G consists now in finding a stopping time $\tau^* : \Omega \to \{1, 2, \ldots, n\}$ such that

$$P[v_{\pi_{\tau^*}} \in \operatorname{Max} G] = \max_{\tau} P[v_{\pi_\tau} \in \operatorname{Max} G],$$

where τ runs over the set of all stopping times with respect to the filtration $(\mathcal{F}_t)_{t=1}^{n}$ and $[v_{\pi_\tau} \in \operatorname{Max} G]$ denotes the set $\{\pi \in \Omega : v_{\pi_\tau} \in \operatorname{Max} G\}$ (we shall use similar notation throughout this paper).

We shall now give an example of possible selector's consecutive observations in the case of a directed path of length 7:

$$(\{1, \ldots, 7\}, (\{(1, 2), (2, 3), \ldots, (6, 7)\}).$$

Let $\pi = (3, 2, 5, 1, 6, 7, 4)$ (see Figure 1).

One can easily notice that only when $t = 1, 3, 5, 6$ does the selector have a chance to make the right choice. One can also notice that if the choice was not made until $t = 6$, there is no chance to get n at the last step. Actually, we can infer from the situation at $t = 6$ that $\pi_7 = 4$.

4. Optimal stopping time. For a fixed $n \in \mathbf{N}$, let us consider a directed path

$$P_n = (\{1, 2, \ldots, n\}, \{(1, 2), \ldots, (n - 1, n)\}).$$

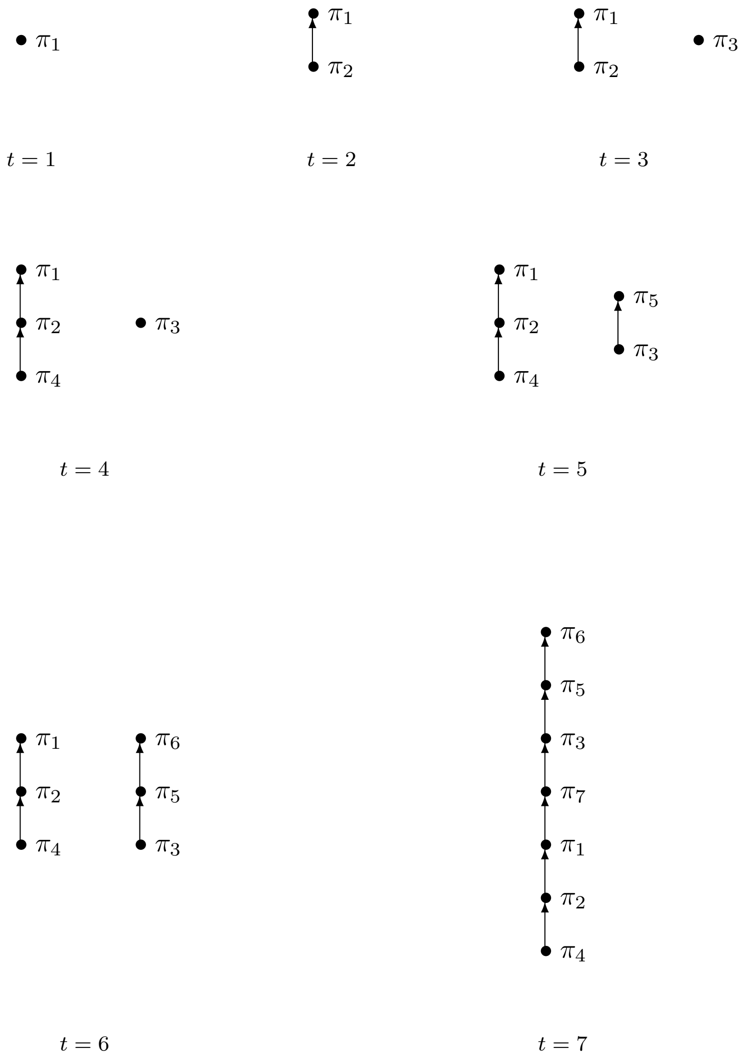
For a subset $A \subseteq \{1, 2, \ldots, n\}$, $c(A)$ is the number of connected components of the graph $(A, \{(1, 2), \ldots, (n - 1, n) \;\cap A^2)$. Within the model defined in the previous section, let

$$\tau^{(n)}(\pi)$$

$$= \min\{t \le n : c(\{\pi_1, \ldots, \pi_t\}) = n - t + 1 \text{ and } \pi_t \in \operatorname{Max} \{\pi_1, \ldots, \pi_t\}\},$$

using the convention $\min \emptyset = n$.

Note that if at time t, $c(\{\pi_1, \ldots, \pi_t\}) = n - t + 1$, then there is no hope that n can be still among the elements to come, namely $\pi_{t+1}, \pi_{t+2}, \ldots, \pi_n$, because we need at least $n - t$ vertices to connect the components we have at time t, and thus all the remaining elements of P_n must be used for this purpose. Therefore the strategy $\tau^{(n)}$ can be described as follows.



$t = 1$ $t = 2$ $t = 3$

$t = 4$ $t = 5$

$t = 6$ $t = 7$

FIG. 1.

Stop when there is a positive conditional (given history) probability that the presently examined candidate is the maximal one and the probability that the maximal one can be among the future candidates is equal to zero.

We are going to prove the following theorem.

THEOREM 4.1. *For a directed path P_n, the stopping time $\tau^{(n)}$ is optimal; i.e.,*

$$P[\pi_{\tau^{(n)}} = n] = \max_{\tau} P[\pi_\tau = n],$$

where τ runs over the set of all stopping times.

We also have

$$P[\pi_{\tau^{(n)}} = n] = \frac{1}{n} \sum_{i=0}^{\left\lfloor \frac{n-1}{2} \right\rfloor} \frac{(n-i-1)\dots(n-2i)}{(n-1)\dots(n-i)}, \tag{1}$$

with the convention that the first term of the above sum is equal to 1.

To prove Theorem 4.1 we shall need Lemma 4.2, stated below.

For a relation $R \subseteq \mathbf{N}^2$, let $A_R^{(m)} = \{\pi \in S_n : (\pi_1, \dots, \pi_m) \cong R\}$.

LEMMA 4.2. *Let* $m \leq n$. *Let* $R \subseteq \mathbf{N}^2$, $\pi \in S_n$, *and* $\pi \in A_R^{(m)}$. *We assume that* $\pi_m \in \operatorname{Max}\{\pi_1, \dots, \pi_m\}$ *(this is, of course, a property of* R*; i.e., for every permutation* $\rho \in A_R^{(m)}$ *we also have* $\rho_m \in \operatorname{Max}\{\rho_1, \dots, \rho_m\}$*). We assume also that for some (or, equivalently, each)* $\pi \in A_R^{(m)}$, *the set* $\{\pi_1, \dots, \pi_m\}$ *has* p *components.*

Then

$$P[\pi_m = n | A_R^{(m)}] = \frac{1}{n - m + 1}. \tag{2}$$

If $m < n$,

$$P[n \in \{\pi_{m+1}, \dots, \pi_n\} | A_R^{(m)}] = \frac{n - p - m + 1}{n - m + 1}. \tag{3}$$

Proof. Each component is a sequence of some consecutive elements from among $1, \dots, m$. Let these components be numbered according to the time of the appearance of their latest elements (the ones that came the last of all the elements of a given component): $C_1, \dots, C_p$.

Assume, losing no generality, that the remaining $n - m$ elements $\pi_{m+1}, \dots, \pi_n$ will appear in the order $\pi_{m+1} < \dots < \pi_n$.

Add to the remaining elements $\pi_{m+1}, \dots, \pi_n$ one dummy element $d > \pi_n$. There are $\binom{n-m+1}{p} p!$ possible choices of different elements $e_1, \dots, e_p$ from among $\pi_{m+1}, \dots, \pi_n, d$ such that e_j is immediately above C_j, $j \leq p$. Now a permutation from $A_R^{(m)}$ is uniquely determined.

Assume that $\pi_m = n$. Thus $m \in C_p$. There are $\binom{n-m}{p-1}(p-1)!$ choices of $p - 1$ different elements $e_1, \dots, e_{p-1}$ from among $\pi_{m+1}, \dots, \pi_n$ such that e_j is immediately above C_j, $j \leq p - 1$. Now a permutation $\pi \in A_R^{(m)}$ such that $\pi_m = n$ is uniquely determined.

Thus

$$P[\pi_m = n | A_R^{(m)}] = \frac{\binom{n-m}{p-1}(p-1)!}{\binom{n-m+1}{p} p!} = \frac{1}{n - m + 1},$$

as required.

As the number of permutations from $A_R^{(m)}$ for which $C_{i_1} < \dots < C_{i_p}$ is the same for all permutations $(i_1, \dots, i_p)$ of $1, \dots, p$, we get

$$P[n \in \{\pi_1, \dots, \pi_m\} | A_R^{(m)}] = P[n \in \bigcup_{i=1}^{p} C_i | A_R^{(m)}]$$

$$= p \cdot P[n \in C_p | A_R^{(m)}] = p \cdot P[n = \pi_m | A_R^{(m)}] = \frac{p}{n - m + 1}.$$

Hence

$$P[n \in \{\pi_{m+1}, \dots, \pi_n\} | A_R^{(m)}] = 1 - \frac{p}{n - m + 1},$$

as required. This completes the proof.

We can now prove Theorem 4.1.

Proof of Theorem 4.1. Let us first make an intuitively obvious, and easy to formalize, observation. Namely, it is sufficient to maximize $P[\pi_\tau = n]$ only over the stopping times τ such that for each $\pi \in S_n$, $\pi_\tau \in \mathrm{Max}\{\pi_1, \dots, \pi_\tau\}$ or $\tau(\pi) = n$.

Now, aiming at contradiction, let us assume that $\tau^{(n)}$ is not optimal. Thus there exists a stopping time τ such that $P[\pi_{\tau^{(n)}} = n] < P[\pi_\tau = n]$. Of course, we can assume that τ is optimal and, according to the remark above, that $\tau(\pi) = t$ only if $\pi_t \in \mathrm{Max}\{\pi_1, \dots, \pi_t\}$ or $t = n$. We can also assume that there is no optimal stopping time $\bar{\tau} \geq \tau$, $\bar{\tau} \neq \tau$.

Let $\tau(\pi) = n$. If $\pi_n = n$, then by the definition of $\tau^{(n)}$ we also have $\tau^{(n)}(\pi) = n$. Thus

$$P[\pi_\tau = n | \tau = n] \leq P[\pi_{\tau^{(n)}} = n | \tau = n].$$

Let us now consider the complementary event $[\tau < n]$.

The event $[\tau < n]$ is a union of disjoint events of the form $A_R^{(m)}$, where $R \subseteq \{1, \dots, m\}^2$ is a relation defined by

$$(i, j) \in R \Leftrightarrow \delta_j = \delta_i + 1, \quad 1 \leq i, j \leq m, m < n,$$

where δ is a permutation such that $\tau(\delta) = m < n$. Thus for at least one $A_R^{(m)}$ we must have

$$P[\pi_\tau = n | A_R^{(m)}] > P[\pi_{\tau^{(n)}} = n | A_R^{(m)}]. \tag{4}$$

Thus on $A_R^{(m)}$ we have $\tau < \tau^{(n)}$.

For τ and R being considered now, let a new stopping time $\bar{\tau}$ be defined by

$$\bar{\tau}(\rho) = \begin{cases} \tau(\rho) \text{ if } \rho \notin A_R^{(m)}, \\ \min M(\rho, m) \text{ if } \rho \in A_R^{(m)} \text{ and } M(\rho, m) \neq \emptyset, \\ n \text{ in the remaining cases} \end{cases}$$

for $\rho \in S_n$, where

$$M(\rho, m) = \{t > m : \rho(t) \in \mathrm{Max}\{\rho_1, \dots, \rho_t\}\}.$$

From (4) and the definition of $\tau^{(n)}$, we infer that $\bar{\tau} \neq \tau$ and $\bar{\tau} \geq \tau$. We are going to show that

$$P[\pi_m \neq n | A_R^{(m)}] \cdot P[\pi_{\bar{\tau}} = n | A_R^{(m)} \cap [\pi_m \neq n]] \geq P[\pi_m = n | A_R^{(m)}]. \tag{5}$$

This implies $P([\pi_{\bar{\tau}} = n] \cap A_R^{(m)} \cap [\pi_m \neq n]) \geq P([\pi_m = n] \cap A_R^{(m)})$ and, thus, $P([\pi_{\bar{\tau}} = n] \cap A_R^{(m)}) \geq P([\pi_\tau = n] \cap A_R^{(m)})$, which gives $P[\pi_{\bar{\tau}} = n] \geq P[\pi_\tau = n]$. This together with $\bar{\tau} \geq \tau$ and $\bar{\tau} \neq \tau$ will contradict our assumption on maximality of τ among all optimal stopping times.

The event that n does not appear at the moment m can be replaced in (5) by the event V that n appears after time m, i.e., $V = [n \in \{\pi_{m+1}, \dots, \pi_n\}]$. Indeed, to prove (5), it is enough to show that

$$P(V | A_R^{(m)}) \cdot P[\pi_{\bar{\tau}} = n | A_R^{(m)} \cap V] \geq P[\pi_m = n | A_R^{(m)}] \tag{6}$$

because

$$P(V|A_R^{(m)}) \cdot P[\pi_{\bar{\tau}} = n|A_R^{(m)} \cap V] = \frac{P(V \cap A_R^{(m)})}{P(A_R^{(m)})} \cdot \frac{P([\pi_{\bar{\tau}} = n] \cap A_R^{(m)} \cap V)}{P(A_R^{(m)} \cap V)}$$

$$= \frac{P([\pi_{\bar{\tau}} = n] \cap A_R^{(m)} \cap [\pi_m \neq n])}{P(A_R^{(m)})},$$

the last equality being a consequence of

$$[\pi_{\bar{\tau}} = n] \cap A_R^{(m)} \cap [\pi_m \neq n] = [\pi_{\bar{\tau}} = n] \cap A_R^{(m)} \cap V = [\pi_{\bar{\tau}} = n] \cap A_R^{(m)}.$$

Let $\bar{A}_R^{(m)} = A_R^{(m)} \cap V$. By (3) of Lemma 4.2 we have

$$P(V|A_R^{(m)}) \cdot P[\pi_{\bar{\tau}} = n|\bar{A}_R^{(m)}]$$

$$= \frac{n-p-m+1}{n-m+1} \sum_{t=m+1}^{n} P([\pi_t = n] \cap [\bar{\tau} = t]|\bar{A}_R^{(m)})$$

$$= \frac{n-p-m+1}{n-m+1} \sum_{t=m+1}^{n} \frac{P([\pi_t = n] \cap [\bar{\tau} = t] \cap \bar{A}_R^{(m)})}{P(\bar{A}_R^{(m)} \cap [\bar{\tau} = t])} \cdot \frac{P([\bar{\tau} = t] \cap \bar{A}_R^{(m)})}{P(\bar{A}_R^{(m)})}$$

$$= \frac{n-p-m+1}{n-m+1} \sum_{t=m+1}^{n} P[\pi_t = n|[\bar{\tau} = t] \cap \bar{A}_R^{(m)}] \cdot P[\bar{\tau} = t|\bar{A}_R^{(m)}].$$

Let us now evaluate $P[\pi_t = n|[\bar{\tau} = t] \cap \bar{A}_R^{(m)}]$ for $t > m$.

Note that for $t > m$, we have at least $p-(t-m)+1$ components of $\{\pi_1, \ldots, \pi_{t-1}\}$, and each of them, maybe with the exception of the first one, is bounded from below by an immediate predecessor that has not appeared by the time t. If $\pi \in [\bar{\tau} = t]$, then none of these predecessors is equal to π_t, because $\pi_t \in \text{Max}\{\pi_1, \ldots, \pi_t\}$. Thus there are at most

$$(n-t+1) - (p-(t-m)) = n-p-m+1$$

elements that can be equal to π_t. As $\pi \in V$ and $\pi \in [\bar{\tau} = t]$, the element n is still among $\{1, \ldots, n\} \setminus \{\pi_1, \ldots, \pi_{t-1}\}$. Hence

$$P[\pi_t = n|[\bar{\tau} = t] \cap \bar{A}_R^{(m)}] \geq \frac{1}{n-p-m+1}.$$

Thus, using (2) of Lemma 4.2 in the last equality below, we finally get

$$P(V|A_R^{(m)}) \cdot P[\pi_{\bar{\tau}} = n|\bar{A}_R^{(m)}]$$

$$\geq \frac{1}{n-m+1} \sum_{t=m+1}^{n} P[\bar{\tau} = t|\bar{A}_R^{(m)}] = \frac{1}{n-m+1} = P[\pi_m = n|A_R^{(m)}].$$

This proves (6) and the optimality of $\tau^{(n)}$.

Now we are going to justify (1).

Let

$$B_j = [\pi_j \in \mathrm{Max}\{\pi_1, \ldots, \pi_j\}],$$

$$C_j = [c(\{\pi_1, \ldots, \pi_j\}) = n - j + 1],$$

$$A_j = B_j \cap C_j.$$

As $C_j = \emptyset$ for $j < \frac{n+1}{2}$, we have

$$P[\pi_{\tau(n)} = n] = \sum_{j=\lceil \frac{n+1}{2} \rceil}^{n} P[\pi_{\tau(n)} = n | A_j] \cdot P(A_j)$$

$$= \sum_{j=\lceil \frac{n+1}{2} \rceil}^{n} P[\pi_{\tau(n)} = n | A_j] \cdot P(B_j | C_j) \cdot P(C_j).$$

We have

$$P(B_j | C_j) = \frac{n - j + 1}{j}$$

and

$$P(C_j) = \frac{\binom{j-1}{n-j} j!(n-j)!}{n!}$$

for $j \geq \frac{n+1}{2}$. Note that $\binom{j-1}{n-j}$ is the number of $0, 1$ sequences $(a_i)_{i=1}^{n}$ such that $a_i = 0$ implies $1 < i < n$, $a_{i-1} = a_{i+1} = 1$, and $|\{i : a_i = 0\}| = n - j$. One can see that this is the case by first forming a sequence of j 1s and then out of all but the first element choosing $n - j$ 1s that will be preceded by 0. The product $j!(n-j)!$ is the number of the permutations in which the i indices in which $a_i = 1$ are at the first j places.

We have

$$P[\pi_{\tau(n)} = n | A_j] = \frac{1}{n - j + 1}$$

because $c(\{\pi_1, \ldots, \pi_j\}) = n - j + 1$. Thus

$$P[\pi_{\tau(n)} = n] = \sum_{j=\lceil \frac{n+1}{2} \rceil}^{n} \frac{1}{j} \binom{j-1}{n-j} j!(n-j)! \frac{1}{n!}$$

or, when summed in reverse order,

$$P[\pi_{\tau(n)} = n] = \sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} \frac{1}{n-i} \binom{n-i-1}{i} (n-i)! i! \frac{1}{n!}$$

$$= \frac{1}{n}\sum_{i=0}^{\lfloor\frac{n-1}{2}\rfloor} \frac{(n-i-1)\dots(n-2i)}{(n-1)\dots(n-i)}.$$

This completes the proof of Theorem 4.1.

We shall now prove a theorem establishing the asymptotics of $P[\pi_{\tau(n)} = n]$. It turns out that $P[\pi_{\tau(n)} = n] = O(\frac{1}{\sqrt{n}})$. Namely, the following theorem holds.

THEOREM 4.3. *We have*

$$\lim_{n\to\infty} P[\pi_{\tau(n)} = n]\sqrt{n} = \frac{\sqrt{\pi}}{2}. \tag{7}$$

Proof. Let $q = 2m$ be a fixed positive even integer. For a positive integer n, let a nonnegative integer k_n satisfy $q^2k_n^4 < n \leq q^2(k_n+1)^4$.

As $\frac{n-1}{2} \geq mqk_n^4 \geq mqk_n^2$, we have

$$\sum_{i=0}^{\lfloor\frac{n-1}{2}\rfloor} \frac{(n-i-1)\dots(n-2i)}{(n-1)\dots(n-i)} > \sum_{i=1}^{mqk_n^2} \frac{(n-i-1)\dots(n-2i)}{(n-1)\dots(n-i)}$$

$$= \sum_{t=1}^{mq} \sum_{j=(t-1)k_n+1}^{tk_n} \sum_{i=(j-1)k_n+1}^{jk_n} \frac{(n-i-1)\dots(n-2i)}{(n-1)\dots(n-i)}$$

$$\geq \sum_{t=1}^{mq} \sum_{j=(t-1)k_n+1}^{tk_n} \sum_{i=(j-1)k_n+1}^{jk_n} \left(\frac{n-2i}{n-i}\right)^{i} \geq k_n \sum_{t=1}^{mq} \sum_{j=(t-1)k_n+1}^{tk_n} \left(\frac{n-2jk_n}{n-jk_n}\right)^{jk_n}$$

$$= k_n \sum_{t=1}^{mq} \sum_{j=(t-1)k_n+1}^{tk_n} \left[\left(1-\frac{jk_n}{n-jk_n}\right)^{n-jk_n-1}\left(1-\frac{jk_n}{n-jk_n}\right)\right]^{\frac{jk_n}{n-jk_n}}$$

$$\geq k_n \sum_{t=1}^{mq} \sum_{j=(t-1)k_n+1}^{tk_n} e^{-\frac{j^2k_n^2}{n-jk_n}} \left(1-\frac{jk_n}{n-jk_n}\right)^{\frac{jk_n}{n-jk_n}}$$

$$\geq k_n \sum_{t=1}^{mq} \left(1-\frac{tk_n^2}{n-tk_n^2}\right)^{\frac{tk_n^2}{n-tk_n^2}} \sum_{j=(t-1)k_n+1}^{tk_n} \left(e^{-\frac{tk_n^3}{n-tk_n^2}}\right)^{j}$$

$$= k_n \sum_{t=1}^{mq} \left(1-\frac{tk_n^2}{n-tk_n^2}\right)^{\frac{tk_n^2}{n-tk_n^2}} \frac{1-\left(e^{-\frac{tk_n^3}{n-tk_n^2}}\right)^{k_n}}{1-e^{-\frac{tk_n^3}{n-tk_n^2}}} \left(e^{-\frac{tk_n^3}{n-tk_n^2}}\right)^{(t-1)k_n+1}.$$

Let

$$A_{n,t} = k_n \left(1-\frac{tk_n^2}{n-tk_n^2}\right)^{\frac{tk_n^2}{n-tk_n^2}} \frac{1-\left(e^{-\frac{tk_n^3}{n-tk_n^2}}\right)^{k_n}}{1-e^{-\frac{tk_n^3}{n-tk_n^2}}} \left(e^{-\frac{tk_n^3}{n-tk_n^2}}\right)^{(t-1)k_n+1}.$$

If $n \to \infty$, then also $k_n \to \infty$. It is not hard to see that

$$\lim_{n\to\infty} \frac{A_{n,t}}{\sqrt{n}} = \frac{q}{t}\left(1 - e^{-\frac{t}{q^2}}\right) e^{-\frac{t(t-1)}{q^2}}.$$

Thus, for every even positive integer $q = 2m$, we have

$$\liminf_{n\to\infty} \frac{1}{\sqrt{n}} \sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} \frac{(n-i-1)\dots(n-2i)}{(n-1)\dots(n-i)} \geq \sum_{t=1}^{mq} \frac{q}{t}\left(1 - e^{-\frac{t}{q^2}}\right) e^{-\frac{t(t-1)}{q^2}}.$$

Estimations similar to those from above can be made if in the series of inequalities above we start with

$$\sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} \frac{(n-i-1)\dots(n-2i)}{(n-1)\dots(n-i)} = \sum_{i=1}^{mqk_n^2} \frac{(n-i-1)\dots(n-2i)}{(n-1)\dots(n-i)} + R(q,n)$$

$$\leq \sum_{t=1}^{mq} \sum_{j=(t-1)k_n+1}^{tk_n} \sum_{i=(j-1)k_n+1}^{jk_n} \left(\frac{n-i-1}{n-1}\right)^i + R(q,n),$$

where $R(q,n)$ denotes the sum of the remaining terms; this term disappears after the limit operation corresponding to the one performed above. Namely, we obtain

$$\limsup_{n\to\infty} \frac{1}{\sqrt{n}} \sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} \frac{(n-i-1)\dots(n-2i)}{(n-1)\dots(n-i)} \leq \frac{1}{q} + \sum_{t=2}^{mq} \frac{q}{t-1}\left(1 - e^{-\frac{t-1}{q^2}}\right) e^{-\frac{(t-1)(t-1)}{q^2}}$$

for every positive even integer $q = 2m$. Applying Lagrange's mean value theorem to the term $e^{-\frac{(t-2)(t-1)}{q^2}} - e^{-\frac{(t-1)(t-1)}{q^2}}$ below, one can also quite easily see that

$$\lim_{q\to\infty} \left(\sum_{t=1}^{mq} \frac{q}{t}\left(1 - e^{-\frac{t}{q^2}}\right) e^{-\frac{t(t-1)}{q^2}} - \sum_{t=2}^{mq} \frac{q}{t-1}\left(1 - e^{-\frac{t-1}{q^2}}\right) e^{-\frac{(t-1)(t-1)}{q^2}} \right)$$

$$= \lim_{q\to\infty} \left(\sum_{t=2}^{mq} \frac{q}{t-1}\left(1 - e^{-\frac{t-1}{q^2}}\right) e^{-\frac{(t-2)(t-1)}{q^2}} - \sum_{t=2}^{mq} \frac{q}{t-1}\left(1 - e^{-\frac{t-1}{q^2}}\right) e^{-\frac{(t-1)(t-1)}{q^2}} \right)$$

$$= \lim_{q\to\infty} \sum_{t=2}^{mq} \frac{q}{t-1}\left(1 - e^{-\frac{t-1}{q^2}}\right) \left(e^{-\frac{(t-1)(t-1)}{q^2}} - e^{-\frac{(t-1)(t-2)}{q^2}} \right) = 0$$

and that, taking only the first term of the Taylor expansion of $1 - e^{-\frac{t-1}{q^2}}$, we also have

$$\lim_{q\to\infty} \left(\sum_{t=2}^{mq} \frac{q}{t-1}\left(1 - e^{-\frac{t-1}{q^2}}\right) e^{-\frac{(t-1)(t-1)}{q^2}} - \sum_{t=1}^{mq} \frac{1}{q} e^{-\frac{(t-1)(t-1)}{q^2}} \right) = 0.$$

Since the expression

$$\sum_{t=1}^{mq} \frac{1}{q} e^{-\frac{(t-1)(t-1)}{q^2}}$$

is a Riemann sum approximating the integral

$$\int_0^\infty e^{-x^2} dx = \frac{\sqrt{\pi}}{2},$$

the proof of Theorem 4.3 is complete.

Acknowledgment. We would like to thank the referees for their remarks that helped us to make this paper shorter and clearer.

REFERENCES

[BG] B. A. Berezovskiy and A. V. Gnedin, *The Problem of Optimal Choice*, Nauka, Moscow, 1984.

[F] T. Ferguson, *Who solved the secretary problem?*, Statist. Sci., 4 (1989), pp. 282–296.

[GM] J. Gilbert and F. Mosteller, *Recognizing the maximum of a sequence*, J. Amer. Statist. Assoc., 61 (1966), pp. 35–73.

[G] A. V. Gnedin, *Multicriteria extensions of the best choice problem: Sequential selection without linear order*, in Stategies for Sequential Search and Selection in Real Time, Contemp. Math. 125, AMS, Providence, RI, 1992, pp. 153–172.

[KMN] M. Kuchta, M. Morayne, and J. Niemiec, *On a universal best choice algorithm for partially ordered sets*, submitted.

[M] M. Morayne, *Partial-order analogue of the secretary problem: The binary tree case*, Discrete Math., 184 (1998), pp. 165–181.

[P] J. Preater, *The best-choice problem for partially ordered sets*, Oper. Res. Lett., 25 (1999), pp. 187–190.

SIAM J. DISCRETE MATH.
Vol. 19, No. 3, pp. 633–650

SIMPLICIAL COMPLEXES OF GRAPHS AND HYPERGRAPHS WITH A BOUNDED COVERING NUMBER[*]

JAKOB JONSSON[†]

Abstract. For $1 \leq p \leq n-1$, define $\mathsf{Cov}_{n,p}$ as the family of graphs on the vertex set $\{1, \ldots, n\}$ with a covering number of at most p (equivalently, with an independence number of at least $n-p$). Since the underlying vertex set is fixed, we may identify each graph in $\mathsf{Cov}_{n,p}$ with its edge set. In particular, we may view $\mathsf{Cov}_{n,p}$ as a simplicial complex. For $i \geq -1$, we show that the rank of the ith homology group of $\mathsf{Cov}_{n,p}$ is a linear combination, with coefficients being polynomials in n, of the ranks of the ith homology groups of $\mathsf{Cov}_{p+2,p}, \ldots, \mathsf{Cov}_{2p+1,p}$. Our proof takes place in a more general setting where we consider complexes of hypergraphs. In addition, we show that the $(2p-1)$-skeleton of $\mathsf{Cov}_{n,p}$ is shellable, which implies that $\mathsf{Cov}_{n,p}$ is $(2p-2)$-connected. For $p \leq 3$, we give a complete description of the homology groups of $\mathsf{Cov}_{n,p}$.

Key words. monotone graph property, simplicial homology, vertex cover, discrete Morse theory

AMS subject classifications. 05E25, 05C69, 06A11, 55U10

DOI. 10.1137/S0895480104443680

1. Introduction.

A (simple) *hypergraph* H consists of a vertex set V and a family E of nonempty subsets of V referred to as *edges*. For a set S of positive integers, H is an *S-hypergraph* if $|e| \in S$ for every $e \in E$. If H is an *$\{r\}$-hypergraph* (i.e., all edges have the same size r), then H is *r-uniform*. A *hypergraph property* is a family of hypergraphs on a fixed vertex set V such that the family is invariant under permutations of V. Since V is fixed, we may identify a given hypergraph with its edge set. A hypergraph property is *monotone* if the property is a simplicial complex (i.e., the property is closed under deletion of edges). Let n, p, r be positive integers. The purpose of this paper is to examine the topology of the following property:

- *H is an r-uniform hypergraph on the vertex set $[n] = \{1, \ldots, n\}$ such that there is a vertex set of size at most p intersecting every edge in H.*

We are particularly interested in the case that all edges have size two, which means that the hypergraphs are ordinary simple graphs.

Formally, we proceed as follows. A hypergraph H is *covered* by a vertex set W if every edge in H contains at least one vertex from W. We refer to W as a *$|W|$-cover* of H. The *covering number* $\tau(H)$ of a hypergraph H is the smallest integer p such that H has a p-cover. For $1 \leq p \leq n$ and $1 \leq r \leq n$, let $\mathsf{Cov}_{n,p,r}$ be the simplicial complex of r-uniform hypergraphs on the vertex set $[n]$ with covering number at most p. The main results of this paper are as follows:

- In sections 6 and 7, we show, for any fixed p and r, that the Betti numbers of $\mathsf{Cov}_{n,p,r}$ over any field $\mathbb{F}$ are polynomials in n. Specifically,

$$\dim \tilde{H}_i(\mathsf{Cov}_{n,p,r}, \mathbb{F}) = \sum_{k=p+r}^{\gamma+1} (-1)^{\gamma+1-k} f_{k,\gamma}(n) \dim \tilde{H}_i(\mathsf{Cov}_{k,p,r}, \mathbb{F}),$$

[*] Received by the editors May 11, 2004; accepted for publication (in revised form) March 28, 2005; published electronically November 15, 2005. This research was partly financed by EC's IHRP Programme, within the Research Training Network "Algebraic Combinatorics in Europe," grant HPRN-CT-2001-00272.

http://www.siam.org/journals/sidma/19-3/44368.html

[†] Department of Mathematics, KTH, SE-10044 Stockholm, Sweden (jakob_jonsson@yahoo.se).

where each $f_{k,\gamma}(n)$ is a polynomial in n and $\gamma = \gamma(p,r)$ is an integer. For $r = 2$, we have that $\gamma = 2p$, which turns out to imply that the degree of $f_{k,\gamma}(n)$ is at most $2p$ in this case.

- In section 8, we give explicit formulas for the homology of $\mathsf{Cov}_{n,p,2}$ for $p \le 3$; for $p = 2$ and $p = 3$, our results are based on computer calculations with Heckenbach's program `homology` [7]. Notably, there is 2-torsion in dimension six in the homology of $\mathsf{Cov}_{n,3,2}$ for $n \ge 6$.
- In section 9, we demonstrate, for any $p \ge 1$, that the $(2p-1)$-skeleton of $\mathsf{Cov}_{n,p,2}$ is vertex-decomposable (see section 3 for definition) and hence shellable. As a consequence, $\mathsf{Cov}_{n,p,2}$ is $(2p-2)$-connected and has no homology in dimension $i \le 2p-2$. For $p \le 3$ and $n \ge 2p+1$, we have detected nonzero homology in dimension $2p-1$. We have not been able to find meaningful counterparts of these results for $\mathsf{Cov}_{n,p,r}$ when $r \ge 3$.

Our results rely heavily on Forman's discrete Morse theory [8]; we give a brief summary of this theory in section 2. In section 5, we introduce a complex $\mathsf{Cov}^{\#}_{n,p,r}$ with the same homotopy type as $\mathsf{Cov}_{n,p,r}$ and with certain nice properties that allow for a smooth analysis. We apply discrete Morse theory to $\mathsf{Cov}^{\#}_{n,p,r}$ in section 6 and derive the polynomial property of the Betti numbers in section 7.

The graph theory presented in section 4 is equally crucial for our theorems and is used throughout the paper. This is classical theory—basically Chapter 13 in Berge [2]—concerning graphs with the property that each vertex is contained in the complement of a cover of minimum size.

1.1. Related work. There are plenty of monotone graph properties in the literature; some examples are complexes of matchings (see Wachs [21]), not k-connected graphs [1, 11, 18, 19, 20], non-Hamiltonian graphs [11], graphs not containing any k-matching [15], and t-colorable graphs [14]. Among all these complexes, the last two seem particularly interesting from our point of view:

- Linusson, Shareshian, and Welker [15] proved that the complex of graphs on n vertices that do not contain any k-matching is homotopy equivalent to a wedge of a certain number of spheres of dimension $3k-4$. For fixed k, this number is a polynomial in n of degree $3k-3$.
- Linusson and Shareshian [14] also examined the complex of t-colorable graphs on n vertices and discovered that all homology is contained in one single dimension for $t = n-2$ and $t = n-3$. Moreover, the nonzero Betti number is a polynomial in n in each of the two cases.

What makes these complexes interesting in the present context is the fact that the Betti numbers are given by polynomials in n, just as for $\mathsf{Cov}_{n,p,r}$. Note that the complexes are defined in terms of cliques and independent sets (which are basically identical concepts, as a clique in a graph is an independent set in the complement of the graph). Namely, one may interpret a k-matching as k disjoint cliques of size 2. Moreover, a graph is t-colorable if and only if it admits a partition of the vertex set into t independent sets. Since the complexes to be examined in this paper are also defined in terms of independent sets (at least for $r = 2$), one may ask whether there is some nice unifying property shared by all these complexes that forces the Betti numbers to be polynomials.

Yet another family of graph properties with Betti numbers given by polynomials appears in the author's thesis [12]. The graphs under consideration have the property that the vertex set is the disjoint union of two independent sets such that one of the sets has size at most some fixed p. This means that the graphs are two-colorable with at most p "black" vertices. In particular, the graphs are bipartite.

1.2. Basic concepts. Let $H = (V, E)$ be a simple hypergraph. We denote the edge $\{a_1, a_2, \dots, a_r\}$ as $a_1 a_2 \cdots a_r$; thus the edge $\{a, b\}$ is denoted as ab. A vertex is *covered* in H if the vertex is contained in some edge in G and *uncovered* otherwise. Whenever the underlying vertex set V is fixed, we identify H with its edge set E; $e \in H$ means that $e \in E$. H is *empty* if $E = \emptyset$ and *nonempty* otherwise. We write $H - e = (V, E \setminus \{e\})$ and $H + e = (V, E \cup \{e\})$. For $W \subset V$, let $H(W)$ be the *induced subhypergraph* of H on the vertex set W obtained by removing $V \setminus W$ along with all edges containing some element from $V \setminus W$. A vertex set U in H is *independent* if no edge in H is a subset of U. The *independence number* $\alpha(H)$ of H is the maximum size of an independent set in H. This means that $\alpha(H) = |V| - \tau(H)$.

A *simplicial complex* on a finite set V is a family of subsets of V closed under deletion of elements. Members of a simplicial complex Σ are called *faces*. The *dimension* of a face σ is defined as $|\sigma| - 1$. The dimension of a complex Σ is the maximal dimension of any face in Σ. A complex is *pure* if all maximal faces have the same dimension. For $d \geq -1$, the *d-simplex* is the simplicial complex of all subsets of a set V of size $d+1$. Note that the (-1)-simplex is the complex containing only the empty set. Whenever we discuss the homology of a simplicial complex, we are referring to the reduced simplicial $\mathbb{Z}$-homology unless otherwise specified. For a simplicial complex Δ and a face σ, the *link* $\mathrm{link}_\Delta(\sigma)$ is the simplicial complex of all $\rho \in \Delta$ such that $\rho \cap \sigma = \emptyset$ and $\rho \cup \sigma \in \Delta$. The *deletion* $\mathrm{del}_\Delta(\sigma)$ is the simplicial complex of all $\rho \in \Delta$ such that $\rho \cap \sigma = \emptyset$.

We obtain the (one-point) *wedge* $\Delta \vee \Gamma$ of two simplicial complexes Δ and Γ—with respect to vertices $x \in \Delta$, $y \in \Gamma$—by taking the disjoint union of Δ and Γ and then identifying x and y. The homotopy type of $\Delta \vee \Gamma$ does not depend on the choice of x and y as long as each of Δ and Γ is (path-) connected. The reduced homology of $\Delta \vee \Gamma$ is the direct sum of the underlying reduced homologies of Δ and Γ.

2. Discrete Morse theory for simplicial complexes.

We give a short review of Forman's discrete Morse theory [8]. Chari [5] and Shareshian [19] have given more elaborate combinatorial interpretations.

Let X be a set and let Δ be a finite family of finite subsets of X. A *matching* on Δ is a family $\mathcal{M}$ of pairs $\{\sigma, \tau\}$ with $\sigma, \tau \in \Delta$ such that no set is contained in more than one pair in $\mathcal{M}$. A set σ in Δ is *critical* or *unmatched* with respect to $\mathcal{M}$ if σ is not contained in any pair in $\mathcal{M}$.

We say that a matching $\mathcal{M}$ on Δ is an *element matching* if every pair in $\mathcal{M}$ is of the form $\{\sigma - x, \sigma + x\}$ for some $x \in X$ and $\sigma \subseteq X$. All matchings considered in this paper are element matchings.

Consider an element matching $\mathcal{M}$ on a family Δ. Let $D = D(\Delta, \mathcal{M})$ be the digraph with vertex set Δ and with a directed edge from σ to τ if and only if either of the following holds:

1. $\{\sigma, \tau\} \in \mathcal{M}$ and $\tau = \sigma + x$ for some $x \notin \sigma$.
2. $\{\sigma, \tau\} \notin \mathcal{M}$ and $\sigma = \tau + x$ for some $x \notin \tau$.

Thus every edge in D corresponds to an edge in the Hasse diagram of Δ ordered by set inclusion; edges corresponding to pairs of matched sets are directed from the smaller set to the larger set, whereas the other edges are directed the other way around. An element matching $\mathcal{M}$ is an *acyclic matching* if D is acyclic, that is, $\sigma \longrightarrow \tau$ and $\tau \longrightarrow \sigma$ implies that $\sigma = \tau$.

An (order-preserving) *poset map* between two posets P and Q is a function $f : P \to Q$ such that $f(x) \leq f(y)$ whenever $x \leq y$. Note that any family of sets has a natural poset structure with order given by set inclusion. In an earlier paper [11], the following simple lemma was provided.

Lemma 2.1 (cluster lemma). *Let $\Delta \subseteq 2^X$ and let $f : \Delta \to Q$ be a poset map, where Q is an arbitrary poset. For $q \in Q$, let $\mathcal{M}_q$ be an acyclic matching on $f^{-1}(q)$. Let*

$$\mathcal{M} = \bigcup_{q \in Q} \mathcal{M}_q;$$

$\mathcal{M}$ is an element matching on Δ. Then $\mathcal{M}$ is an acyclic matching on Δ.

Remark. Hersh [10] discovered Lemma 2.1 independently of our work. Björner (personal communication) suggested the formulation in terms of poset maps.

For the remainder of this section, Δ is a simplicial complex; for convenience, we assume that $\Delta \supsetneqq \{\emptyset\}$. Given an acyclic matching $\mathcal{M}$ on Δ, we may without loss of generality assume that the empty set $\emptyset$ is contained in some pair in $\mathcal{M}$. Namely, if all sets of size 1 are matched with larger sets, then there is obviously a cycle in $D(\Delta, \mathcal{M})$.

Theorem 2.2 (see Forman [8]). *Let Δ be a simplicial complex and let $\mathcal{M}$ be an acyclic matching on Δ such that the empty set is not critical. Then Δ is homotopy equivalent to a cell complex with one cell of dimension $p \geq 0$ for each critical face of dimension p in Δ plus one additional 0-cell.*

We write $\sigma \longrightarrow \tau$ if there is a directed path from σ to τ in $D(\Delta, \mathcal{M})$. For families $\mathcal{V}$ and $\mathcal{W}$, we write $\mathcal{V} \longrightarrow \mathcal{W}$ if there are $V \in \mathcal{V}$ and $W \in \mathcal{W}$ such that $V \longrightarrow W$. The symbol $\not\longrightarrow$ is used to denote the nonexistence of such a directed path. Let $\mathcal{U} = \mathcal{U}(\Delta, \mathcal{M})$ be the family of critical faces in Δ with respect to $\mathcal{M}$. For a (possibly empty) family $\mathcal{V} \subseteq \mathcal{U}$, let

$$\Delta_{\mathcal{V}} = \{\sigma \in \Delta : \mathcal{V} \longrightarrow \sigma\} \cup \{\emptyset, \{x\}\}, \tag{2.1}$$

where $\{x\}$ is the set matched with the empty set in $\mathcal{M}$. If $\mathcal{V}$ is nonempty, then $\Delta_{\mathcal{V}} = \{\sigma \in \Delta : \mathcal{V} \longrightarrow \sigma\}$.

Lemma 2.3 (see [11]). *$\Delta_{\mathcal{V}}$ is a subcomplex of Δ and $\mathcal{U}(\Delta_{\mathcal{V}}, \mathcal{M}_{\mathcal{V}}) = \Delta_{\mathcal{V}} \cap \mathcal{U}(\Delta, \mathcal{M})$, where $\mathcal{M}_{\mathcal{V}}$ is the restriction of $\mathcal{M}$ to $\Delta_{\mathcal{V}}$.*

Theorem 2.4 (see [11]). *Suppose that $\mathcal{V} \subseteq \mathcal{U} = \mathcal{U}(\Delta, \mathcal{M})$ has the property that $\mathcal{U} \setminus \mathcal{V} \not\longrightarrow \mathcal{V}$ and $\mathcal{V} \not\longrightarrow \mathcal{U} \setminus \mathcal{V}$. Then Δ is homotopy equivalent to $\Delta_{\mathcal{V}} \vee \Delta_{\mathcal{U} \setminus \mathcal{V}}$. In particular, Δ is homotopy equivalent to $\Delta_{\mathcal{U}}$. More generally, if $\mathcal{U}$ is the disjoint union of families $\mathcal{V}_1, \ldots, \mathcal{V}_r$ with the property that $\mathcal{V}_i \not\longrightarrow \mathcal{V}_j$ if $i \neq j$, then Δ is homotopy equivalent to $\bigvee_{i=1}^{r} \Delta_{\mathcal{V}_i}$.*

We obtain the latter statement in Theorem 2.4 from the former statement via a simple induction argument.

3. Vertex-decomposable complexes.

In this section, we review the basics on vertex-decomposable complexes and present some elementary properties of vertex-decomposable skeletons of simplicial complexes.

Definition 3.1. *We define the class of* vertex-decomposable *(VD) simplicial complexes recursively as follows:*

(i) *The void complex $\emptyset$, the (-1)-simplex $\{\emptyset\}$, and any 0-simplex $\{\emptyset, \{v\}\}$ are VD.*

(ii) *If Δ is pure and contains a vertex x—a* shedding vertex*—such that* $\mathrm{del}_\Delta(x)$ *and* $\mathrm{link}_\Delta(x)$ *are VD, then Δ is also VD.*

Provan and Billera [17] introduced VD complexes.

Theorem 3.2 (see Provan and Billera [17]). *VD complexes are shellable. As a consequence, a d-dimensional VD complex is $(d-1)$-connected.*

Lemma 3.3. *Let Δ be simplicial complex and let v be a vertex in Δ. If the $(d-1)$-skeleton of* $\mathrm{link}_\Delta(v)$ *and the d-skeleton of* $\mathrm{del}_\Delta(v)$ *are VD, then the d-skeleton of Δ is VD.*

The following simple lemma is used in the proof of Theorem 9.1.

Lemma 3.4. *Let $\Delta_1, \dots, \Delta_k$ be simplicial complexes and let $d_1, \dots, d_k$ be integers such that the d_i-skeleton of Δ_i is VD for each i; $d_i \geq -1$. Then the $(\sum_i d_i + k - 1)$-skeleton of the join $\Delta_1 * \cdots * \Delta_k$ is VD.*

Proof. Use double induction on the size of $\Delta_1 * \cdots * \Delta_k$ and $d = \sum_i d_i + k - 1$. If $d = -1$, then we are done. Otherwise, let i be such that $d_i \geq 0$; say that $i = k$. Let v be a shedding vertex for the d_k-skeleton Σ_k of Δ_k; $\mathrm{link}_{\Sigma_k}(v)$ and $\mathrm{del}_{\Sigma_k}(v)$ are VD. Write $\Delta = \Delta_1 * \cdots * \Delta_{k-1}$.

If v is a cone point in Σ_k, then $\Sigma_k = \Delta_k$, and so the (d_k-1)-skeleton of $\mathrm{del}_{\Delta_k}(v) = \mathrm{link}_{\Delta_k}(v)$ coincides with $\mathrm{del}_{\Sigma_k}(v) = \mathrm{link}_{\Sigma_k}(v)$. By induction, $\Delta * \mathrm{link}_{\Delta_k}(v)$ has a VD $(d-1)$-skeleton. $\Delta * \Delta_k$ is the cone over this complex and hence has a VD d-skeleton.

If v is not a cone point in Σ_k, then the d_k-skeleton of $\mathrm{del}_{\Delta_k}(v)$ and the $(d_k - 1)$-skeleton of $\mathrm{link}_{\Delta_k}(v)$ are VD. By induction, $\Delta * \mathrm{del}_{\Delta_k}(v)$ has a VD d-skeleton, whereas $\Delta * \mathrm{link}_{\Delta_k}(v)$ has a VD $(d-1)$-skeleton. By Lemma 3.3, we obtain that the d-skeleton of Δ is VD, and we are done. □

4. Solid hypergraphs. Let us say that a hypergraph $H = (V, E)$ with covering number p is (p,r)-*solid* if, for every vertex set U of size at most $r-1$, there is a p-cover W of H such that $U \cap W = \emptyset$. In this section, we present some useful results about (p,r)-solid $[r]$-hypergraphs; recall that a hypergraph is an S-hypergraph if all edges are of size an integer in S.

Lemma 4.1. *If an $[r]$-hypergraph H is (p,r)-solid, then H is r-uniform. Moreover, every covered vertex is contained in a p-cover of H.*

Proof. For the first statement, since H is (p,r)-solid, a vertex set of size at most $r-1$ cannot form an edge in H. For the second statement, let v be a covered vertex and let e be an edge in H containing v; clearly, $|e \setminus \{v\}| = r-1$. H being (p,r)-solid means that some p-cover does not intersect $e \setminus \{v\}$. Since this cover must then contain v, we are done. □

By Lemma 4.1, we may restrict our attention to r-uniform hypergraphs. First, we make a simple observation.

Lemma 4.2. *If H is r-uniform and has covering number p, then the number of vertices in H is at least $p + r - 1$. In particular, this is true if H is (p,r)-solid.*

Proof. Any r-uniform hypergraph on at most $p + r - 2$ vertices has a covering number of at most $p - 1$. □

The bound in Lemma 4.2 is tight; the complete r-uniform hypergraph on $p+r-1$ vertices is (p,r)-solid.

We will use the following lemma in section 7 to prove that the Betti numbers of $\mathsf{Cov}_{n,p,r}$ are polynomials in n for each fixed p and r.

Lemma 4.3. *For every $p, r \geq 1$, there is a positive integer $\gamma(p,r)$ such that if H is a (p,r)-solid and r-uniform hypergraph with no uncovered vertices, then the number of vertices in H is at most $\gamma(p,r)$.*

Proof. Let H be (p,r)-solid without uncovered vertices. If we remove an edge e such that $\tau(H) = \tau(H - e)$, then $H - e$ is again (p,r)-solid with no uncovered vertices. Namely, assume to the contrary that some vertex $v \in e$ is uncovered in $H - e$. By Lemma 4.1, there is a p-cover W of H containing v. However, this implies that $W \setminus \{v\}$ is a $(p-1)$-cover of $H - e$, which is a contradiction.

Starting with H, remove edges not affecting the covering number until we have a τ-*critical* hypergraph H', meaning that the removal of any edge in H' decreases the covering number of H' (this is equivalent to H' being α-*critical* as defined by Berge [2, sec. 13.3]). By a result of Bollobás [4], the number of edges in a τ-critical r-uniform hypergraph with covering number p is at most $\binom{p+r-1}{r}$; see Lovász [16, Ex. 13.32]. This implies that the number of vertices in H' is at most $r \cdot \binom{p+r-1}{r}$ (this is a very loose bound), and the lemma follows. □

For $r = 2$, we can establish a tight bound on $\gamma(p, r)$.

THEOREM 4.4 (see Berge [2, Thm. 13.13]). *If G is a simple graph with $\tau(G) = p$ such that G contains no uncovered vertices and such that every vertex is contained in a p-cover, then the number of vertices in G is at most $2p$. As a consequence, if G is $(p, 2)$-solid with no uncovered vertices, then the number of vertices in G is at most $2p$.*

The bound $2p$ is tight, as the $2p$-cycle is $(p, 2)$-solid. The first statement in the theorem is basically a consequence of some results due to Hajnal [9]; see Berge [2, Thm. 13.8–9]. Unfortunately, these results seem hard to generalize to hypergraphs. By Lemma 4.1, the second statement in the theorem is a consequence of the first.

Finally, we state and prove a few results that we will use in section 9 to prove that the $(2p-1)$-skeleton of $\mathsf{Cov}_{n,p,2}$ is VD; hence we restrict our attention to graphs.

LEMMA 4.5. *Let H be a graph with covering number p and with connected components $C_1, \dots, C_k$. Then H is $(p, 2)$-solid if and only if there are integers $p_1, \dots, p_k$ summing up to p such that C_i is $(p_i, 2)$-solid for each i.*

Proof. With $p_i = \tau(C_i)$, it is clear that $\sum_i p_i = \tau(H) = p$. Suppose that some vertex $v \in C_i$ is contained in every p_i-cover of C_i. Then v is contained in every p-cover of H; we cannot cover $H \setminus C_i$ with fewer than $p - p_i$ vertices. Conversely, if v is *not* contained in a given p_i-cover of C_i, then we can extend this cover to a p-cover of H not containing v by picking an arbitrary p_j-cover of every other C_j. □

LEMMA 4.6. *A $(p, 2)$-solid graph H contains at least $2p - k$ edges, where k is the number of connected components in H with at least two vertices.*

Proof. The lemma is clear for $p = 1$; assume that $p \geq 2$. We may assume that H contains no uncovered vertices. Let the connected components of H be $C_1, \dots, C_k$. With $p_i = \tau(C_i)$, we have that C_i is $(p_i, 2)$-solid for each i by Lemma 4.5. In particular, if $k \geq 2$, then we may use induction on p to conclude that C_i contains at least $2p_i - 1$ edges. Summing over i and using the fact that $\sum_i p_i = p$, we obtain that H contains at least $2p - k$ edges.

Thus assume that H is connected. As in the proof of Lemma 4.3, note that if we remove an edge that does not affect the covering number of H, then the resulting graph is again $(p, 2)$-solid with no uncovered vertices. Remove such edges from H until we have a τ-critical graph H'; the removal of any edge from H' decreases the covering number.

If the obtained graph H' is disconnected with k components, then we remove at least $k - 1$ edges, and by the same induction argument as above, H' contains at least $2p - k$ edges. Hence H contains at least $2p - 1$ edges as desired.

Assume that H' is connected; for simplicity, let us write H instead of H'. Berge [2, Thm. 13.6] proved that a τ-critical and connected graph is 2-connected. We want to find a vertex x in H such that the induced subgraph K obtained by removing x from H is $(p-1, 2)$-solid. By induction, this will imply that K contains at least $2(p-1)-1$ edges, which in turn will imply that H contains at least $2(p-1) - 1 + 2 = 2p - 1$ edges as desired. Namely, we get rid of at least two edges when we remove x, and the resulting graph K is connected, as H is 2-connected.

To find the vertex x, let $y \leq z$ mean that any p-cover of H containing y also contains z. This defines a partial order. Namely, since H is τ-critical, we have, for each y, w such that $yw \in H$, that the graph $H - yw$ has a $(p-1)$-cover Q with the property that $y, w \notin Q$. If $z \notin Q$, then $y \not\leq z$, as $Q \cup \{y\}$ is a p-cover of H not containing z. If $z \in Q$, then $z \not\leq y$, as $Q \cup \{w\}$ is a p-cover of H containing z but not y. Now, pick x maximal with respect to the given partial order. This means, for any $y \neq x$, that there is a p-cover of H containing x but not y. In particular, there is a $(p-1)$-cover not containing y of the induced subgraph K obtained by removing x from H. However, this means that K is $(p-1, 2)$-solid, and we are done. □

The bound in Lemma 4.6 is tight: Let G be the graph consisting of a path of vertex length $2(p-k+1)$ and $k-1$ additional components, each of vertex size two. Then G is $(p, 2)$-solid and contains $k - 1 + 2p - 2k + 1 = 2p - k$ edges.

5. A related simplicial complex. For $n, p, r \geq 1$, let $\mathsf{Cov}^{\#}_{n,p,r}$ be the simplicial complex of $[r]$-hypergraphs on the vertex set $[n]$ with covering number at most p. Hence $\mathsf{Cov}^{\#}_{n,p,r}$ consists of hypergraphs with edges of size between 1 and r, whereas $\mathsf{Cov}_{n,p,r}$ consists of r-uniform hypergraphs. As it turns out, $\mathsf{Cov}^{\#}_{n,p,r}$ has several attractive properties that make the complex easier to handle than the original $\mathsf{Cov}_{n,p,r}$.

LEMMA 5.1. *For $1 \leq p \leq n$ and $1 \leq r \leq n$, $\mathsf{Cov}_{n,p,r} \simeq \mathsf{Cov}^{\#}_{n,p,r}$.*

Proof. We show how to collapse $\mathsf{Cov}^{\#}_{n,p,r}$ down to $\mathsf{Cov}_{n,p,r}$. Fix a linear order on $\binom{[n]}{r}$; this is the family of edges of maximum size r. For a hypergraph $H \in \mathsf{Cov}^{\#}_{n,p,r} \setminus \mathsf{Cov}_{n,p,r}$, let $e = e(H)$ be maximal with respect to this linear order such that e contains an edge $e' \in H$ of size at most $r-1$; e itself is not necessarily contained in H. For each e of size r, let $\mathcal{F}(e)$ be the family of hypergraphs $H \in \mathsf{Cov}^{\#}_{n,p,r} \setminus \mathsf{Cov}_{n,p,r}$ such that $e(H) = e$. It is clear that the families $\mathcal{F}(e)$ satisfy the Cluster Lemma 2.1. Namely, $H \mapsto e(H) \in \binom{[n]}{r}$ is a poset map with the given linear order on $\binom{[n]}{r}$. Now, we obtain a perfect matching on $\mathcal{F}(e)$ by pairing $H + e$ with $H - e$ for each $H \in \mathcal{F}(e)$. Namely, adding or deleting e does not affect $e(H)$. Also, the covering number remains the same when e is added or deleted, as H already contains an edge $e' \subsetneqq e$. By the Cluster Lemma 2.1, we are done. □

Next, we prove that $\mathsf{Cov}^{\#}_{n,p,r}$ and $\mathsf{Cov}^{\#}_{n,r,p}$ are homotopy equivalent; we may hence swap p and r without affecting the homotopy type. For this, we will need the following special case of the Nerve Theorem.

THEOREM 5.2 (see Björner [3]). *For a given simplicial complex Δ, let the* nerve *$\mathsf{N}(\Delta)$ of Δ be the simplicial complex with one vertex for each maximal face in Δ and with $\{\sigma_i : i \in I\}$ a face in $\mathsf{N}(\Delta)$ if and only if the intersection $\bigcap_{i \in I} \sigma_i$ is nonempty. Then Δ and $\mathsf{N}(\Delta)$ are homotopy equivalent.*

PROPOSITION 5.3. *For $n, p, r \geq 1$, we have that $\mathsf{Cov}^{\#}_{n,p,r} \simeq \mathsf{Cov}^{\#}_{n,r,p}$. In particular, $\mathsf{Cov}_{n,p,r} \simeq \mathsf{Cov}_{n,r,p}$ whenever $n \geq \max\{p, r\}$.*

Proof. For $1 \leq n \leq p + r - 1$, $\mathsf{Cov}^{\#}_{n,p,r}$ and $\mathsf{Cov}^{\#}_{n,r,p}$ are both cones and hence collapsible; every edge of maximum size is a cone point. Assume that $n \geq p + r$. Consider the nerve complex $\mathsf{N}_{n,p,r} = \mathsf{N}(\mathsf{Cov}^{\#}_{n,p,r})$. We may identify the vertices in $\mathsf{N}_{n,p,r}$ with subsets of $[n]$ of size p. Namely, every maximal hypergraph $H \in \mathsf{Cov}^{\#}_{n,p,r}$ has a unique p-cover consisting of those x with the property that the singleton edge x belongs to H.

For a set U of size p, let H_U be the maximal hypergraph in $\mathsf{Cov}^{\#}_{n,p,r}$ with unique p-cover U. A family $\mathcal{W}$ of vertices in $\mathsf{N}_{n,p,r}$ forms a face of $\mathsf{N}_{n,p,r}$ if and only if the intersection $\bigcap_{W \in \mathcal{W}} H_W$ is nonempty. This means that there is a set S of size at most r such that $|W \cap S| \geq 1$ for each $W \in \mathcal{W}$. However, this is exactly the condition that the hypergraph $([n], \mathcal{W})$ admits a cover of size at most r. As a consequence, we may identify $\mathsf{N}_{n,p,r}$ with $\mathsf{Cov}_{n,r,p}$. Thus

$$\mathsf{Cov}^{\#}_{n,p,r} \simeq \mathsf{N}_{n,p,r} \cong \mathsf{Cov}_{n,r,p} \simeq \mathsf{Cov}^{\#}_{n,r,p};$$

the first equivalence follows from Theorem 5.2, whereas the last equivalence follows from Lemma 5.1. □

6. An acyclic matching. The purpose of this section is to present an acyclic matching on $\mathsf{Cov}^{\#}_{n,p,r}$ such that the unmatched graphs have certain rather strong properties. Observant readers may note that our matching is quite similar in nature to the matching that Linusson and Shareshian [14, sec. 5] provided for complexes of t-colorable graphs; see section 2 for information about the underlying discrete Morse theory.

For an $[r]$-hypergraph H on the vertex set $[n]$, let $\mathcal{X}(H)$ be the family of all subsets of $[n-1]$ of size at most $r-1$ that have nonempty intersection with every p-cover of $H([n-1])$. Note that if $H \in \mathsf{Cov}^{\#}_{n,p,r}$, then we may add the edge $X \cup \{n\}$ to H for any $X \in \mathcal{X}(H)$ without ending up outside $\mathsf{Cov}^{\#}_{n,p,r}$.

Define

$$\begin{aligned} \mathcal{A}_{n,p,r} &= \{H \in \mathsf{Cov}^{\#}_{n,p,r} : H([n-1]) \in \mathsf{Cov}^{\#}_{n-1,p-1,r}\}; \\ \mathcal{B}_{n,p,r} &= \{H \in \mathsf{Cov}^{\#}_{n,p,r} : H([n-1]) \notin \mathsf{Cov}^{\#}_{n-1,p-1,r} \text{ and } \mathcal{X}(H) \neq \emptyset\}; \\ \mathcal{C}_{n,p,r} &= \{H \in \mathsf{Cov}^{\#}_{n,p,r} : H([n-1]) \notin \mathsf{Cov}^{\#}_{n-1,p-1,r} \text{ and } \mathcal{X}(H) = \emptyset\}. \end{aligned}$$

It is clear that $\mathsf{Cov}^{\#}_{n,p,r}$ is the disjoint union of $\mathcal{A}_{n,p,r}$, $\mathcal{B}_{n,p,r}$, and $\mathcal{C}_{n,p,r}$, and that $\mathcal{A}_{n,p,r}$ and $\mathcal{A}_{n,p,r} \cup \mathcal{C}_{n,p,r}$ are both simplicial complexes. This implies that the three families satisfy the Cluster Lemma 2.1. We want to prove that there are perfect acyclic matchings on $\mathcal{A}_{n,p,r}$ and $\mathcal{B}_{n,p,r}$. The remaining family $\mathcal{C}_{n,p,r}$ is the family of all $[r]$-hypergraphs H such that $H([n-1])$ has covering number p and such that every subset of $[n-1]$ of size at most $r-1$ is disjoint from some p-cover of $H([n-1])$. This means that $H([n-1])$ is (p,r)-solid.

We obtain a perfect acyclic matching on $\mathcal{A}_{n,p,r}$ by pairing $H - n$ with $H + n$; we match with the singleton edge n. Namely, for any cover W of $H([n-1])$, $W \cup \{n\}$ is a cover of H.

For a family $\mathcal{X}$ of subsets of $[n-1]$, let $\mathcal{B}_{n,p,r}(\mathcal{X})$ be the family of hypergraphs $H \in \mathcal{B}_{n,p,r}$ such that $\mathcal{X}(H) = \mathcal{X}$. It is clear that the families $\mathcal{B}_{n,p,r}(\mathcal{X})$ satisfy the Cluster Lemma 2.1. Namely, $H \mapsto \mathcal{X}(H)$ is a poset map; $\mathcal{X}(H)$ cannot grow when we delete edges from H. Let $X(H)$ be minimal in $\mathcal{X}(H)$ with respect to some fixed linear order. If $H \in \mathcal{B}_{n,p,r}(\mathcal{X})$, then the same is true for $H + X(H)n$; every p-cover of H contains an element from $X(H)$, and $X(H)$ has size at most $r-1$. $\mathcal{X}(H)$ does not depend on the set of edges containing n, which means that we obtain a perfect matching on $\mathcal{B}_{n,p,r}(\mathcal{X})$ by pairing $H - X(H)n$ with $H + X(H)n$. Taking the union over all $\mathcal{X}$, we get a perfect acyclic matching on $\mathcal{B}_{n,p,r}$.

Combining our two perfect acyclic matchings on $\mathcal{A}_{n,p,r}$ and $\mathcal{B}_{n,p,r}$, we obtain an acyclic matching on $\mathsf{Cov}^{\#}_{n,p,r}$ with $\mathcal{C}_{n,p,r}$ as the set of critical graphs. Theorem 2.4 yields the following proposition.

Proposition 6.1. *With notation as above and as in* (2.1) *in section* 2*,*

$$\mathsf{Cov}^{\#}_{n,p,r} \simeq (\mathsf{Cov}^{\#}_{n,p,r})_{\mathcal{C}_{n,p,r}}.$$

Also, given an acyclic matching on $\mathcal{C}_{n,p,r}$ with c_i critical sets of dimension i for each i, $\mathsf{Cov}^{\#}_{n,p,r}$ is homotopy equivalent to a cell complex with c_i cells of dimension i for each i and one additional 0*-cell.*

7. Homotopy type and homology. Before proceeding, let us examine some special cases. First of all, note that $\mathsf{Cov}^{\#}_{n,p,r}$ is a cone and hence collapsible whenever $1 \le n \le p+r-1$. Also, by Lemma 5.1, $\mathsf{Cov}^{\#}_{p+r,p,r}$ is homotopy equivalent to $\mathsf{Cov}_{p+r,p,r}$, which contains all r-uniform hypergraphs on the vertex set $[p+r]$ except the complete hypergraph. This implies that

$$\mathsf{Cov}^{\#}_{p+r,p,r} \simeq S^{C(p+r,r)-2}, \tag{7.1}$$

where $C(m,k) = \binom{m}{k}$.

Next, consider $p = 1$; the complex $\mathsf{Cov}^{\#}_{n,1,r}$ consists of *star hypergraphs*, which are hypergraphs covered by a single vertex. By Proposition 5.3, $\mathsf{Cov}^{\#}_{n,1,r}$ is homotopy equivalent to $\mathsf{Cov}^{\#}_{n,r,1} = \mathsf{Cov}_{n,r,1}$. Now, the latter complex is obviously the $(r-1)$-skeleton of an $(n-1)$-simplex. As a consequence, we have the following simple result.

PROPOSITION 7.1. *For $n, r \ge 1$, $\mathsf{Cov}^{\#}_{n,1,r}$ and $\mathsf{Cov}^{\#}_{n,r,1}$ are both homotopy equivalent to a wedge of $\binom{n-1}{r}$ spheres of dimension $r-1$.*

Now, proceed with general n, p, r. Recall that $\mathcal{C}_{n,p,r}$ is the set of critical hypergraphs in Proposition 6.1 and that a hypergraph H in $\mathsf{Cov}^{\#}_{n,p,r}$ belongs to $\mathcal{C}_{n,p,r}$ if and only if $H([n-1])$ is (p,r)-solid. For a nonempty vertex set $J \subseteq [n-1]$, let $\mathcal{C}_{n,p,r}(J)$ be the family of hypergraphs H in $\mathcal{C}_{n,p,r}$ such that J is the set of vertices that are covered in $H([n-1])$. Write

$$\begin{aligned} \Lambda_{n,p,r}(J) &= (\mathsf{Cov}^{\#}_{n,p,r})_{\mathcal{C}_{n,p,r}(J)} \quad \text{(notation as in (2.1))};\\ \Lambda_{k,p,r} &= \Lambda_{k,p,r}([k-1]). \end{aligned}$$

LEMMA 7.2. *Let $n, p, r \ge 1$. For any nonempty vertex set $J \subseteq [n-1]$, $\Lambda_{n,p,r}(J)$ is the union of $\mathcal{C}_{n,p,r}(J)$ and a collapsible subcomplex of the complex $\mathcal{A}_{n,p,r}$ defined in section* 6. *Moreover, we have that*

$$\Lambda_{n,p,r}(J) \simeq \Lambda_{|J|+1,p,r}. \tag{7.2}$$

Proof. For the first claim, let H be a hypergraph in $\mathcal{C}_{n,p,r}(J)$. We want to prove that $H \not\longrightarrow \mathcal{C}_{n,p,r}(I)$ if $I \neq J$. Note that if we remove an edge e from H, then we obtain a hypergraph in $\mathcal{C}_{n,p,r}(I)$ for some $I \subseteq J$ or a hypergraph in $\mathcal{A}_{n,p,r}$. If $n \in e$, then $H - e \in \mathcal{C}_{n,p,r}(J)$; thus assume that $n \notin e$. It is clear that $\mathcal{A}_{n,p,r} \not\longrightarrow \mathcal{C}_{n,p,r}$, which means that we only have to prove that if the new hypergraph $G = H-e$ belongs to $\mathcal{C}_{n,p,r}(I)$, then $I = J$.

Assume the opposite. Then some $x \in e$ is uncovered in $G([n-1])$. Since $H \in \mathcal{C}_{n,p,r}$, there is a p-cover W of $H([n-1])$ such that $(e \setminus \{x\}) \cap W = \emptyset$. Since $e \in H([n-1])$, we must have that $x \in W$. However, since x is uncovered in $G([n-1])$, $W \setminus \{x\}$ covers $G([n-1])$, which implies that $G \in \mathcal{A}_{n,p,r}$, contradictory to assumption. Thus our claim is proved.

For the second claim, we have that the first claim implies that $\Lambda_{n,p,r}(J)$ is the union of $\mathcal{C}_{n,p,r}(J)$ and a collapsible subcomplex $\mathcal{T}$ of $\mathcal{A}_{n,p,r}$. To see that $\mathcal{T}$ is collapsible, just note that $H-n \in \mathcal{T}$ if and only if $H+n \in \mathcal{T}$; this is by definition of $\Lambda_{n,p,r}(J)$ and Lemma 2.3. In particular, $\mathcal{T}$ is a cone with the singleton edge n forming a cone point. $\mathcal{C}_{n,p,r}(J) \cup \mathcal{T}$ is easily seen to be homotopy equivalent to $\mathcal{C}_{n,p,r}(J) \cup \mathcal{A}_{n,p,r}$. Namely, we obtain a perfect acyclic matching on $\mathcal{A}_{n,p,r} \setminus \mathcal{T}$ by pairing $H-n$ with $H+n$ whenever $H \in \mathcal{A}_{n,p,r} \setminus \mathcal{T}$; $\mathcal{A}_{n,p,r}$ and $\mathcal{T}$ are both cones with cone point n.

Let $\mathcal{C}'_{n,p,r}(J)$ be the subfamily of $\mathcal{C}_{n,p,r}(J)$ consisting of those H with the property that all vertices in $[n-1] \setminus J$ are uncovered in H (not only in $H([n-1])$). We obtain a perfect acyclic matching on $\mathcal{C}_{n,p,r}(J) \setminus \mathcal{C}'_{n,p,r}(J)$ in the following manner. In a hypergraph $H \in \mathcal{C}_{n,p,r}(J) \setminus \mathcal{C}'_{n,p,r}(J)$, define $e(H)$ as the maximal edge in H with respect to some fixed linear order such that $e(H)$ contains some vertex in $[n-1] \setminus J$. Let $\mathcal{C}_{n,p,r}(J,e)$ be the subfamily of $\mathcal{C}_{n,p,r}(J) \setminus \mathcal{C}'_{n,p,r}(J)$ consisting of those H satisfying $e(H) = e$.

It is clear that the families $\mathcal{C}_{n,p,r}(J,e)$ satisfy the Cluster Lemma 2.1. Namely, $H \mapsto e(H)$ is a poset map; $e(H)$ cannot increase when edges are removed from H. Write $e'(H) = (e(H) \cap J) \cup \{n\}$. We claim that we may define a perfect matching on $\mathcal{C}_{n,p,r}(J,e)$ by pairing $H - e'(H)$ with $H + e'(H)$ whenever $H \in \mathcal{C}_{n,p,r}(J,e)$; note that $e'(H)$ is the same for all $H \in \mathcal{C}_{n,p,r}(J,e)$. To prove the claim, it suffices to prove that $H - e'(H) \in \mathcal{C}_{n,p,r}(J)$ if and only if $H + e'(H) \in \mathcal{C}_{n,p,r}(J)$; $e(H)$ does not depend on whether the edge $e'(H)$ is present in H. To prove this, we need only show that $H + e'(H) \in \mathsf{Cov}^{\#}_{n,p,r}$ whenever $H \in \mathcal{C}_{n,p,r}(J)$. Now, every p-cover W of H is contained in J by assumption; otherwise, we would have a $(p-1)$-cover of $H([n-1])$. This implies that W must contain an element from $e(H) \cap J = e'(H) \setminus \{n\}$. Thus W intersects e', and we are done.

The conclusion is that the simplicial complex $\mathcal{C}_{n,p,r}(J) \cup \mathcal{A}_{n,p,r}$ is homotopy equivalent to $\mathcal{C}'_{n,p,r}(J) \cup \mathcal{A}_{n,p,r}$. Now, $\mathcal{C}'_{n,p,r}(J) \cup \mathcal{A}_{n,p,r}(J)$ is a simplicial complex, where $\mathcal{A}_{n,p,r}(J)$ is the set of all graphs in $\mathcal{A}_{n,p,r}$ such that all vertices in $[n-1] \setminus J$ are uncovered. We may collapse $\mathcal{C}'_{n,p,r}(J) \cup \mathcal{A}_{n,p,r}$ down to $\mathcal{C}'_{n,p,r}(J) \cup \mathcal{A}_{n,p,r}(J)$ by matching $H - n$ with $H + n$ whenever $H \in \mathcal{A}_{n,p,r} \setminus \mathcal{A}_{n,p,r}(J)$. The resulting complex is clearly isomorphic to $\mathcal{C}_{|J|+1,p,r}([|J|]) \cup \mathcal{A}_{|J|+1,p,r}$. By the proof above, we may collapse this complex down to $\Lambda_{|J|+1,p,r}$, and we are done. □

Define

$$\gamma(p,r) = \min\{\gamma : \mathcal{C}_{n,p,r}(J) = \emptyset \text{ whenever } |J| > \gamma\}. \tag{7.3}$$

Such a $\gamma(p,r)$ exists by Lemma 4.3, and $\gamma(p,2) = 2p$ by Theorem 4.4.

THEOREM 7.3. *Let $n,p,r \geq 1$. With notation as above,*

$$\mathsf{Cov}^{\#}_{n,p,r} \simeq \bigvee_{k=p+r}^{\gamma(p,r)+1} \bigvee_{\binom{n-1}{k-1}} \Lambda_{k,p,r} = \bigvee_{k=p+r}^{\min\{\gamma(p,r)+1,n\}} \bigvee_{\binom{n-1}{k-1}} \Lambda_{k,p,r}, \tag{7.4}$$

where $\gamma = \gamma(p,r)$ is defined as in (7.3)*; $\gamma(p,r) = pr$ for $1 \leq r \leq 2$.*

Remark. Since the 1-skeleton of $\mathsf{Cov}^{\#}_{n,p,r}$ is full as soon as $p \geq 2$, the right-hand side in (7.4) is unambiguous from a homotopy point of view. For $p = 1$, $\mathsf{Cov}^{\#}_{n,p,r}$ is homotopy equivalent to a wedge of spheres in a fixed dimension by Proposition 7.1, which immediately yields unambiguity.

Proof. First, note that Lemma 7.2 implies that

$$\mathsf{Cov}^{\#}_{n,p,r} \simeq \bigvee_{J \subseteq [n-1]} \Lambda_{n,p,r}(J) \simeq \bigvee_{k=1}^{n} \bigvee_{\binom{n-1}{k-1}} \Lambda_{k,p,r}.$$

Namely, by the proof of the lemma, $\mathcal{C}_{n,p,r}(J) \not\longrightarrow \mathcal{C}_{n,p,r}(I)$ if $I \neq J$; hence Theorem 2.4 yields the desired result. To settle the theorem, it remains to prove that $\mathcal{C}_{n,p,r}(J)$ is empty unless $p + r \leq |J| + 1 \leq \gamma(p,r) + 1$. The lower bound follows by Lemma 4.2, whereas the upper bound is by definition of $\gamma(p,r)$; see (7.3). □

COROLLARY 7.4. *Let $p,r \geq 1$ and $n \geq \gamma(p,r)+1$. For any field $\mathbb{F}$ and any integer $i \geq -1$, $\tilde{H}_i(\mathsf{Cov}^{\#}_{n,p,r}, \mathbb{F})$ is nonzero if and only if $\tilde{H}_i(\mathsf{Cov}^{\#}_{\gamma(p,r)+1,p,r}, \mathbb{F})$ is nonzero.*

Moreover, the connectivity degrees of the complexes $\mathsf{Cov}^{\#}_{n,p,r}$ *and* $\mathsf{Cov}^{\#}_{\gamma(p,r)+1,p,r}$ *are the same. In particular, for* $n \geq 2p+1$, $\tilde{H}_i(\mathsf{Cov}^{\#}_{n,p,2}, \mathbb{F})$ *is nonzero if and only if* $\tilde{H}_i(\mathsf{Cov}^{\#}_{2p+1,p,2}, \mathbb{F})$ *is nonzero, and the connectivity degrees of* $\mathsf{Cov}^{\#}_{n,p,2}$ *and* $\mathsf{Cov}^{\#}_{2p+1,p,2}$ *are the same.*

Proof. Whenever $n \geq \gamma(p,r)+1$, $\tilde{H}_i(\mathsf{Cov}^{\#}_{n,p,r}, \mathbb{F})$ is nonzero if and only if $\tilde{H}_i(\Lambda_{k,p,r}, \mathbb{F})$ is nonzero for some k such that $p+r \leq k \leq \gamma(p,r)+1$; use Theorem 7.3. By the same theorem, the connectivity degree of $\mathsf{Cov}^{\#}_{n,p,r}$ is the minimum of the connectivity degrees of $\Lambda_{k,p,r}$ for $p+r \leq k \leq \gamma(p,r)+1$. Since these conditions do not depend on n, we are done. For the last claim, apply Theorem 4.4. □

Proposition 7.5 (folklore). *Let* $d \geq 0$ *and let* f *be a polynomial of degree at most* d. *Then, for any* $s \in \mathbb{Z}$ *and all* $x \in \mathbb{C}$,

$$f(x) = \sum_{k=s}^{d+s} (-1)^{d+s-k} \binom{x-s}{k-s} \binom{x-1-k}{d+s-k} f(k);$$

the binomial coefficients are interpreted as polynomials in the natural manner.

Proof. It is easily checked that the left-hand and right-hand sides coincide for $x = s, 1+s, \dots, d+s$. A polynomial of degree at most d is uniquely determined by its values on any $d+1$ points, which concludes the proof. □

Corollary 7.6. *Let* $n,p,r \geq 1$. *For any field* $\mathbb{F}$ *and any integer* $i \geq -1$, *the Betti number* $\beta_i(\mathsf{Cov}^{\#}_{n,p,r}, \mathbb{F}) = \dim \tilde{H}_i(\mathsf{Cov}^{\#}_{n,p,r}, \mathbb{F})$ *satisfies*

$$\beta_i(\mathsf{Cov}^{\#}_{n,p,r}, \mathbb{F}) = \sum_{k=p+r}^{\gamma+1} (-1)^{\gamma+1-k} \binom{n-1}{k-1} \binom{n-1-k}{\gamma+1-k} \beta_i(\mathsf{Cov}^{\#}_{k,p,r}, \mathbb{F});$$

$\gamma = \gamma(p,r)$. *In particular,* $\beta_i(\mathsf{Cov}^{\#}_{n,p,r}, \mathbb{F})$ *is a polynomial in* n *of degree at most* $\gamma(p,r)$.

Remark. Since $\gamma(p,2) = 2p$ by Theorem 4.4, we have that

$$\beta_i(\mathsf{Cov}^{\#}_{n,p,2}, \mathbb{F}) = \sum_{k=p+2}^{2p+1} (-1)^{k-1} \binom{n-1}{k-1} \binom{n-1-k}{2p+1-k} \beta_i(\mathsf{Cov}^{\#}_{k,p,2}, \mathbb{F}).$$

By Proposition 5.3, we may choose $\gamma(p,r) = pr$ in the corollary whenever $p \leq 2$.

Proof. By Theorem 7.3, we know that $f_{p,r,i}(n) = \beta_i(\mathsf{Cov}^{\#}_{n,p,r}, \mathbb{F})$ defines a polynomial in n of degree at most $\gamma(p,r)$ such that $f_{p,r,i}(k) = 0$ for $1 \leq k \leq p+r-1$. By Proposition 7.5 with $s = 1$, we are done. □

For the remainder of this section, we confine ourselves to the case $r = 2$.

Corollary 7.7. *Let* $\mathbb{F}$ *be a field or* $\mathbb{Z}$. *For* $1 \leq p \leq n-2$, $\tilde{H}_i(\mathsf{Cov}_{n,p,2}, \mathbb{F}) = \tilde{H}_i(\mathsf{Cov}^{\#}_{n,p,2}, \mathbb{F})$ *is zero unless* $i \leq p \cdot \min\{p+1, \frac{n+1}{2}\} - 1$. *Hence, for* $2 \leq q \leq n-1$, $\tilde{H}_i(\mathsf{Cov}_{n,n-q,2}, \mathbb{F})$ *is zero unless* $i \leq \lfloor \frac{(n+1)(n-q)}{2} \rfloor - 1$, *which implies that the Alexander dual of* $\mathsf{Cov}_{n,n-q,2}$ *has no homology strictly below dimension* $\lceil \frac{(q-2)(n+1)}{2} \rceil - 1$.

Proof. It is clear that all hypergraphs $G \in \mathcal{C}_{k,p,2}([k-1])$ are ordinary graphs; since $G([k-1])$ is $(p,2)$-solid, $G([k-1])$ has this property (apply Lemma 4.1), and the singleton edge k cannot be present in G. We claim that a graph $G \in \mathcal{C}_{k,p,2}([k-1])$ has at most $p \cdot \frac{k+1}{2}$ edges; inserting $k = \min\{2p+1, n\}$ yields the desired bound. Now, by construction, the degree of each vertex in $G([k-1])$ is at most p; otherwise some vertices would necessarily be part of every p-cover of $G([k-1])$. Also, the vertex k is not part of any p-cover, which implies that the degree of k is at most p. Summing, we get $p \cdot \frac{k-1}{2} + p = p \cdot \frac{k+1}{2}$ as claimed. The last statement follows by Alexander duality; $\binom{n}{2} - (\frac{(n+1)(n-q)}{2} - 1) - 3 = \frac{(q-2)(n+1)}{2} - 1$. □

Remark. In section 9, we show that $\tilde{H}_i(\mathsf{Cov}_{n,p,2})$ is zero unless $i \geq 2p - 1$.

The last statement in Corollary 7.7 looks a bit similar to a result of Linusson and Shareshian [14], which states that the complex Col_n^t of t-colorable graphs on n vertices is $(\lceil \frac{(t-1)(n-1)\{2}{\rceil} - 2)$-connected. In this context, it might be worth noting that Col_n^t is contained in the Alexander dual of $\mathsf{Cov}_{n,n-(t+1),2}$; a t-colorable graph does not contain any $(t+1)$-cliques. Since our acyclic matching is closely related to the acyclic matching of Linusson and Shareshian [14], it is therefore not too surprising that our bound is only slightly different from theirs; see section 11 for a potential improvement of this bound.

Finally, we prove a minor result about the reduced Euler characteristic $\tilde{\chi}(\mathsf{Cov}_{n,p,2}^{\#})$ of $\mathsf{Cov}_{n,p,2}^{\#}$. Note that Corollaries 7.6 and 7.7 imply that $\tilde{\chi}(\mathsf{Cov}_{n,p,2}^{\#})$ defines a polynomial in n of degree at most $2p$ for each fixed p.

Proposition 7.8. *Let $p \geq 1$ and let f_p be the polynomial with the property that $f_p(n) = \tilde{\chi}(\mathsf{Cov}_{n,p,2}^{\#})$ for $n \geq 1$. Then $f_p(0) = -1$. Moreover, let $\mathcal{Y}_{n,p}$ be the family of hypergraphs in $\mathsf{Cov}_{n,p,2}^{\#}$ with no uncovered vertices. Then $\tilde{\chi}(\mathcal{Y}_{n,p}) = 0$ whenever $n > 2p$, where $\tilde{\chi}(\mathcal{Y}_{n,p}) = -\sum_{Y \in \mathcal{Y}_{n,p}} (-1)^{|Y|}$.*

Proof. Define $\mathsf{Cov}_{0,p,2}^{\#} = \mathcal{Y}_{0,p} = \{\emptyset\}$. Clearly,

$$\tilde{\chi}(\mathsf{Cov}_{n,p,2}^{\#}) = \sum_{k=0}^{n} \binom{n}{k} \tilde{\chi}(\mathcal{Y}_{k,p}) \tag{7.5}$$

for all $n \geq 0$. Moreover, for $n \geq 1$,

$$\tilde{\chi}(\mathsf{Cov}_{n,p,2}^{\#}) = f_p(n) = \sum_{k \geq 0} \binom{n}{k} y_k, \tag{7.6}$$

where $y_k = 0$ for $k > 2p$; the degree of f_p is at most $2p$. One easily derives from (7.5) and (7.6) that

$$y_n - \tilde{\chi}(\mathcal{Y}_{n,p}) = (-1)^n (y_0 - \tilde{\chi}(\mathcal{Y}_{0,p})) = (-1)^n (y_0 + 1)$$

for $n \geq 0$. Thus it suffices to prove that $\tilde{\chi}(\mathcal{Y}_{n,p}) = 0$ for some $n > 2p$; this will imply that $y_0 = \tilde{\chi}(\mathcal{Y}_{0,p}) = -1$ and hence that $f_p(0) = \tilde{\chi}(\mathsf{Cov}_{0,p,2}^{\#}) = -1$ as desired. As a byproduct, we will also obtain that $\tilde{\chi}(\mathcal{Y}_{n,p}) = 0$ for all $n > 2p$.

Now, for a given hypergraph $H \in \mathcal{Y}_{n,p}$, let H^* be the graph obtained from H by removing all singleton edges. Let $\mathcal{X}_{n,p}$ be the subfamily of $\mathcal{Y}_{n,p}$ consisting of all hypergraphs H such that some vertex x is contained in every p-cover of the underlying graph H^*. For each $H \in \mathcal{X}_{n,p}$, let $x(H)$ be minimal with this property. We obtain a perfect element matching on $\mathcal{X}_{n,p}$ by pairing $H - \{x(H)\}$ and $H + \{x(H)\}$.

Let $H \in \mathcal{Y}_{n,p} \setminus \mathcal{X}_{n,p}$ and let W be a p-cover of H. By assumption, for each $w \in W$, there is a p-cover of H^* not containing w, which implies that w is adjacent to at most p vertices in H. It follows that there are at most $p + p^2$ covered vertices in H; hence $\mathcal{Y}_{n,p} \setminus \mathcal{X}_{n,p} = \emptyset$ whenever $n > p + p^2$. As a consequence, $\tilde{\chi}(\mathcal{Y}_{n,p}) = 0$ whenever $n > p + p^2$, and we are done. □

8. Computations. Corollary 7.6 reduces the problem of determining the homology of $\mathsf{Cov}_{n,p,r} \simeq \mathsf{Cov}^{\#}_{n,p,r}$ for general $n \geq p+r$ to the special cases $p+r \leq n \leq \gamma(p,r)+1$. For $r=2$, we know by Theorem 4.4 that it suffices to consider $p+2 \leq n \leq 2p+1$. Using the computer program `homology` [7], we have been able to compute the homology of $\mathsf{Cov}_{n,p} = \mathsf{Cov}_{n,p,2}$ for $p = 2, 3$; the results are presented in Theorems 8.1 and 8.2.

For integers m, r, define $C(m,r) = \binom{m}{r}$.

THEOREM 8.1. *For $n \geq 4$, the kth homology group of $\mathsf{Cov}_{n,2}$ is zero unless $3 \leq k \leq 4$, in which case we have that*

$$\begin{aligned} \tilde{H}_3(\mathsf{Cov}_{n,2}) &\cong \mathbb{Z}^{C(n-1,4)}; \\ \tilde{H}_4(\mathsf{Cov}_{n,2}) &\cong \mathbb{Z}^{C(n,4)}. \end{aligned}$$

In particular, the reduced Euler characteristic of $\mathsf{Cov}_{n,2}$ is $\binom{n-1}{3}$.

Proof. Running `homology` [7] on the complex $\mathsf{Cov}_{5,2}$, we obtain that

$$\begin{aligned} \tilde{H}_3(\mathsf{Cov}_{5,2}) &\cong \mathbb{Z}; \\ \tilde{H}_4(\mathsf{Cov}_{5,2}) &\cong \mathbb{Z}^5. \end{aligned}$$

By (7.1), $\mathsf{Cov}_{4,2} \simeq S^4$. Thus Corollary 7.6 yields that the homology of $\mathsf{Cov}_{n,2}$ is torsion-free and that

$$\begin{aligned} &\dim \tilde{H}_3(\mathsf{Cov}_{n,2}, \mathbb{Q}) = \tbinom{n-1}{5-1}\tbinom{n-5-1}{4-5+1} = \tbinom{n-1}{4}; \\ &\dim \tilde{H}_4(\mathsf{Cov}_{n,2}, \mathbb{Q}) = -\tbinom{n-1}{4-1}\tbinom{n-4-1}{4-4+1} + 5\tbinom{n-1}{5-1}\tbinom{n-5-1}{4-5+1} = \tbinom{n}{4}. \qquad \square \end{aligned}$$

Remark. We have not been able to determine the homotopy type of $\mathsf{Cov}_{n,2}$.

THEOREM 8.2. *For $n \geq 5$, the kth homology group of $\mathsf{Cov}_{n,3}$ is zero unless $5 \leq k \leq 8$, in which case we have that*

$$\begin{aligned} \tilde{H}_5(\mathsf{Cov}_{n,3}) &\cong \mathbb{Z}^{C(n-1,6)}; \\ \tilde{H}_6(\mathsf{Cov}_{n,3}) &\cong (\mathbb{Z}_2)^{C(n,6)}; \\ \tilde{H}_7(\mathsf{Cov}_{n,3}) &\cong \mathbb{Z}^{9C(n,6)}; \\ \tilde{H}_8(\mathsf{Cov}_{n,3}) &\cong \mathbb{Z}^{C(n,5)}. \end{aligned}$$

In particular, the reduced Euler characteristic of $\mathsf{Cov}_{n,3}$ is $-\binom{n-1}{4} \cdot \frac{5n^2-31n+15}{15}$. By Proposition 5.3, the same holds for the complex $\mathsf{Cov}_{n,2,3}$.

Proof. Computations with `homology` [7] yield that

$$\begin{cases} \tilde{H}_6(\mathsf{Cov}_{6,3}) \cong \mathbb{Z}_2; \\ \tilde{H}_7(\mathsf{Cov}_{6,3}) \cong \mathbb{Z}^9; \\ \tilde{H}_8(\mathsf{Cov}_{6,3}) \cong \mathbb{Z}^6 \end{cases} \quad \text{and} \quad \begin{cases} \tilde{H}_5(\mathsf{Cov}_{7,3}) \cong \mathbb{Z}; \\ \tilde{H}_6(\mathsf{Cov}_{7,3}) \cong (\mathbb{Z}_2)^7; \\ \tilde{H}_7(\mathsf{Cov}_{7,3}) \cong \mathbb{Z}^{63}; \\ \tilde{H}_8(\mathsf{Cov}_{7,3}) \cong \mathbb{Z}^{21}. \end{cases}$$

By (7.1), we know that $\tilde{H}_i(\mathsf{Cov}_{5,3}) = \mathbb{Z}$ if $i = 8$ and 0 otherwise. By Corollary 7.6, there is no torsion in $\tilde{H}_i(\mathsf{Cov}_{n,3}, \mathbb{Z})$ unless $i = 6$, in which case there is 2-torsion but no free homology. Corollary 7.6 yields that

$$\begin{aligned} &\dim \tilde{H}_5(\mathsf{Cov}_{n,3}, \mathbb{Q}) = \tbinom{n-1}{6}\tbinom{n-8}{0} = \tbinom{n-1}{6}; \\ &\dim \tilde{H}_6(\mathsf{Cov}_{n,3}, \mathbb{Z}_2) = -\tbinom{n-1}{5}\tbinom{n-7}{1} + 7\tbinom{n-1}{6}\tbinom{n-8}{0} = \tbinom{n}{6}; \\ &\dim \tilde{H}_7(\mathsf{Cov}_{n,3}, \mathbb{Q}) = -9\tbinom{n-1}{5}\tbinom{n-7}{1} + 63\tbinom{n-1}{6}\tbinom{n-8}{0} = 9\tbinom{n}{6}; \\ &\dim \tilde{H}_8(\mathsf{Cov}_{n,3}, \mathbb{Q}) = \tbinom{n-1}{4}\tbinom{n-6}{2} - 6\tbinom{n-1}{5}\tbinom{n-7}{1} + 21\tbinom{n-1}{6}\tbinom{n-8}{0} = \tbinom{n}{5}. \qquad \square \end{aligned}$$

Table 1

The homology of $\Lambda_{k,p,2}$ *for all interesting* (k,p) *such that* $2 \leq p \leq 3$ *and for* $(k,p) = (6,4), (7,4)$ *(we obtained the latter homology via a computer calculation of the homology of* $\mathsf{Cov}_{7,4}$*).*

$\tilde{H}_i(\Lambda_{k,p,2}, \mathbb{Z})$	$i = 3$	4	5	6	7	8	9	10	11	12	13
$(k,p) = (4,2)$	-	$\mathbb{Z}$	-	-	-	-	-	-	-	-	-
$(5,2)$	$\mathbb{Z}$	$\mathbb{Z}$	-	-	-	-	-	-	-	-	-
$(5,3)$	-	-	-	-	-	$\mathbb{Z}$	-	-	-	-	-
$(6,3)$	-	-	-	$\mathbb{Z}_2$	$\mathbb{Z}^9$	$\mathbb{Z}$	-	-	-	-	-
$(7,3)$	-	-	$\mathbb{Z}$	$\mathbb{Z}_2$	$\mathbb{Z}^9$	-	-	-	-	-	-
$(6,4)$	-	-	-	-	-	-	-	-	-	-	$\mathbb{Z}$
$(7,4)$	-	-	-	-	-	-	-	$\mathbb{Z}$	$\mathbb{Z}^{55} \oplus \mathbb{Z}_2$	-	$\mathbb{Z}$

Remark. Note that all Betti numbers are integer multiples of binomial coefficients. This is due to Theorem 7.3 and the simple structure of the homology of $\Lambda_{k,p,2}$; see Table 1. We would be surprised if this property held in general; see Proposition 11.1 (e) for a potential conjecture that might be a bit more realistic.

9. Connectivity degree of $\mathsf{Cov}_{n,p,2}$. We prove a result about the connectivity degree of $\mathsf{Cov}_{n,p} = \mathsf{Cov}_{n,p,2}$. Specifically, we prove that $\mathsf{Cov}_{n,p}$ has a vertex-decomposable (VD) $(2p-1)$-skeleton; see section 3 for definition. Note that we consider the graph complex $\mathsf{Cov}_{n,p}$, not the hypergraph complex $\mathsf{Cov}^{\#}_{n,p,2}$ We have not been able to prove anything of interest about the connectivity degree of $\mathsf{Cov}_{n,p,r}$ for $r \geq 3$.

For a simplicial complex Δ and disjoint vertex sets I and E, we define

$$\Delta(I,E) = \{\sigma : I \cup \sigma \in \Delta, (I \cup E) \cap \sigma = \emptyset\} = \mathrm{link}_{\mathrm{del}_{\Delta}(E)}(I).$$

Theorem 9.1. *For* $1 \leq p \leq n-2$, *the* $(2p-1)$*-skeleton of* $\mathsf{Cov}_{n,p}$ *is* VD. *In particular,* $\mathsf{Cov}_{n,p}$ *is* $(2p-2)$*-connected.*

Proof. Let $Y = \binom{[n-1]}{2}$ and $E_n = \{1n, \ldots, (n-1)n\}$. For any disjoint subsets A, B of Y, let

$$d_p(A,B) = p + \min\{p-1, |Y \setminus B|\} - |A|;$$

$d_p(A,B) = 2p - 1 - |A|$ if $|Y \setminus B| \geq p-1$. We claim that the $d_p(A,B)$-skeleton of $\mathsf{Cov}_{n,p}(A,B)$ is VD. The special case $A = B = \emptyset$ yields the theorem, since $|Y| \geq p-1$.

To prove the claim, we use induction on $|Y \setminus B|$. We distinguish three cases:

(i) $|Y \setminus B| \leq p-1$. Then the covering number of the graph with edge set $Y \setminus B$ is at most $p-1$. As a consequence, the graph with edge set $E_n \cup (Y \setminus B)$ has covering number at most p, which implies that all edges in $E_n \cup (Y \setminus (A \cup B))$ are cone points in $\mathsf{Cov}_{n,p}(A,B)$. In particular, $\mathsf{Cov}_{n,p}(A,B)$ is the full simplex on

$$|E_n| + |Y \setminus B| - |A| = n - 1 + |Y \setminus B| - |A| \geq d_p(A,B) + 1$$

elements $(n-1 \geq p+1)$. This implies that the $d_p(A,B)$-skeleton of $\mathsf{Cov}_{n,p}(A,B)$ is VD as desired.

(ii) $|Y \setminus B| \geq p$ and $A \subsetneq Y \setminus B$. Then let $e \in Y \setminus (A \cup B)$. We have by induction on $|Y \setminus (A \cup B)|$ that the link $\mathsf{Cov}_{n,p}(A+e,B)$ has a VD $(2p-2-|A|)$-skeleton and that the deletion $\mathsf{Cov}_{n,p}(A,B+e)$ has a VD $(2p-1-|A|)$-skeleton. As a consequence, the $(2p-1-|A|)$-skeleton of $\mathsf{Cov}_{n,p}(A,B)$ is VD; use Lemma 3.3.

(iii) $|Y \setminus B| = |A| \geq p$ and $A = Y \setminus B$. In this case, we consider complexes $\mathsf{Cov}_{n,p}(A, Y \setminus A)$ such that $|A| \geq p$. Note that all faces of $\mathsf{Cov}_{n,p}(A, Y \setminus A)$ are subsets of E_n. Let H be the graph with edge set A. We identify three subcases:

(a) $\tau(H) \leq p - 1$. Then all $n - 1$ edges in E_n are cone points in $\mathsf{Cov}_{n,p}(A, B)$, and we are done; $|E_n| - 1 = n - 2 \geq p \geq 2p - |A| > d_p(A, Y \setminus A)$.

(b) $\tau(H) = p$ and some vertex x is contained in every p-cover of H. Then the edge xn is a cone point in $\mathsf{Cov}_{n,p}(A, B)$. In particular, the $(2p - 1 - |A|)$-skeleton of $\mathsf{Cov}_{n,p}(A, Y \setminus A)$ is VD if and only if the $(2p - 2 - |A|)$-skeleton of $\mathsf{Cov}_{n,p}(A + xn, Y \setminus A)$ is VD.

Define A_0 and Y_0 as the sets obtained from A and Y by removing all edges containing x; hence $Y_0 = \binom{[n-1]\setminus\{x\}}{2}$. We have that $\mathsf{Cov}_{n,p}(A + xn, Y \setminus A)$ coincides with $\mathsf{Cov}_{n-1,p-1}(A_0, Y_0 \setminus A_0)$, where we remove the vertex x (rather than n) to obtain $\mathsf{Cov}_{n-1,p-1}$. Namely, a graph G containing H and being contained in $H + E_n$ has a p-cover if and only if $G([n] \setminus \{x\})$ has a $(p - 1)$-cover. By induction on n, the d_{p-1}-skeleton of $\mathsf{Cov}_{n-1,p-1}(A_0, Y_0 \setminus A_0)$ is VD, where $d_{p-1} = d_{p-1}(A_0, Y_0 \setminus A_0)$.

We need to prove that

$$d_{p-1} \geq 2p - 2 - |A|.$$

Now,

$$d_{p-1} = p - 1 + \min\{p - 2, |A_0|\} - |A_0|.$$

If $p - 2 \geq |A_0|$, then $d_{p-1} = p - 1$, which is at least $2p - 1 - |A|$, as $|A| \geq p$. If $p - 2 < |A_0|$, then $d_{p-1} = 2p - 3 - |A_0|$, which is at least $2p - 2 - |A|$ as $|A \setminus A_0| \geq 1$. In fact, we must have $|A \setminus A_0| > 1$, because x is contained in every p-cover. Thus we are done.

(c) $\tau(H) = p$ and no vertex is contained in every p-cover of H. This means that H is $(p, 2)$-solid. As a consequence, Lemma 4.6 yields that $|A| \geq 2p - k$, where k is the number of connected components of H with at least two vertices. Thus it suffices to prove that $\Delta = \mathsf{Cov}_{n,p}(A, Y \setminus A)$ has a VD $(k - 1)$-skeleton. Let $C_1, \dots, C_k$ be the connected components of H(uncovered vertices excluded); by Lemma 4.5, each C_i is $(p_i, 2)$-solid for some $p_i \geq 1$ satisfying $\sum_i p_i = p$. Let T_i be the set of edges $xn \in E_n$ with one endpoint x in C_i. Let Δ_i be the induced subcomplex of Δ on the set T_i. It is clear that $\Delta = \Delta_1 * \cdots * \Delta_k$; we can add a subset Q of E_n to H without increasing $p = \tau(H)$ if and only if we can add the corresponding subsets $Q \cap T_i$ without increasing $p_i = \tau(C_i)$.

Now, each vertex in C_i is contained in a p_i-cover of C_i by Lemma 4.1, and $p_i \geq 1$ for each i. As a consequence, the 0-skeleton of Δ_i is VD for each i, which implies by Lemma 3.4 that the $(k - 1)$-skeleton of Δ is VD. Thus we are done. □

We conjecture that there is homology in dimension $2p - 1$ for $n \geq 2p + 1$; this would imply that $\mathsf{Cov}_{n,p}$ is *not* $(2p-1)$-connected in general; see section 11 for further discussion.

10. Triangle-free graphs. Note that $\mathsf{Cov}_{n,p}$ is the Alexander dual of the complex of graphs on n vertices that do not contain a clique of size $n - p$. For $p = n - 3$, we obtain the complex $\not\triangleright_n$ of triangle-free graphs on n vertices. In this section, we summarize our humble results for this very important graph property.

Table 2

The homology of $\ntriangleright_n$ for $4 \leq n \leq 7$. The figures are collected from Table 1 and translated via Alexander duality.

$\tilde{H}_i(\ntriangleright_n, \mathbb{Z})$	$i = 2$	3	4	5	6	7	8
$n = 4$	$\mathbb{Z}^3$	-	-	-	-	-	-
5	-	$\mathbb{Z}^5$	$\mathbb{Z}$	-	-	-	-
6	-	-	$\mathbb{Z}^6$	$\mathbb{Z}^9 \oplus \mathbb{Z}_2$	-	-	-
7	-	-	-	$\mathbb{Z}^7$	$\mathbb{Z}_2$	$\mathbb{Z}^{55}$	$\mathbb{Z}$

Corollary 10.1. *For $n \geq r + 2$,*

$$\mathsf{Cov}_{n,n-r-1,r} \simeq \Lambda_{n,n-r-1,r} \vee \bigvee_{n-1} \Lambda_{n-1,n-r-1,r} \simeq \Lambda_{n,n-r-1,r} \vee \bigvee_{n-1} S^{C(n-1,r)-2},$$

where $C(m,k) = \binom{m}{k}$. In particular, for $r = 2$, the dual complex $\ntriangleright_n$ of triangle-free graphs on n vertices has the property that $\tilde{H}_{n-2}(\ntriangleright_n, \mathbb{Z})$ contains $\mathbb{Z}^{n-1}$ as a free subgroup.

Proof. The first equivalence is Theorem 7.3. The second equivalence follows from the fact that

$$\Lambda_{p+r,p,r} \simeq \mathsf{Cov}_{p+r,p,r} \simeq S^{C(p+r,r)-2};$$

use Theorem 7.3 and (7.1) with $p = n - r - 1$. For the final statement, use Alexander duality. □

We have no complete description of the homology of $\ntriangleright_n$ except for $n \leq 7$; see Table 2. However, we know that there is no homology below dimension $n - 2$.

Proposition 10.2 (see Jonsson [12]). *For $n \geq 2$, the $(n-2)$-skeleton of $\ntriangleright_n$ is VD; hence $\ntriangleright_n$ is $(n-3)$-connected.*

11. Concluding remarks and open problems. We have not been able to compute the homology of $\mathsf{Cov}_{n,p,r}$ for general n, p, and r, and we have very little hope to ever see this being achieved; see the complexity-theoretic remark below for some further discussion. Nevertheless, the homology of $\mathsf{Cov}_{n,p} = \mathsf{Cov}_{n,p,2}$ certainly has plenty of structure, and our computations for small values of n and p suggest that there is quite some more structure to be found. In the following proposition, note that we restrict our attention to $p \leq 3$.

Proposition 11.1. *The following hold for $1 \leq p \leq 3$:*

(a) *$\mathsf{Cov}_{n,p}$ has no homology over any field strictly above dimension $\binom{p+2}{2} - 2$. Equivalently, the Alexander dual of $\mathsf{Cov}_{n,n-r}$ has no homology strictly below dimension $d_{n,r} = n(r-2) - \binom{r-1}{2} - 1$.*

(b) *For $p+2 \leq n \leq 2p+1$, $\mathsf{Cov}_{n,p}$ has no homology strictly below dimension $2p-1+\binom{2p-n+2}{2}$.*

(c) *For $p = 2$ and $p = 3$, $\tilde{H}_{\binom{p+2}{2}-2}(\mathsf{Cov}_{n,p}, \mathbb{Z})$ is free of rank $\binom{n}{p+2}$.*

(d) *$\tilde{H}_{2p-1}(\mathsf{Cov}_{n,p}, \mathbb{Z})$ is free of rank $\binom{n-1}{2p}$.*

(e) *For $i \geq 2p$ and for any field $\mathbb{F}$,*

$$\sum_{k=p+2}^{2p+1} (-1)^k \beta_i(\Lambda_{k,p,2}, \mathbb{F}) = 0.$$

Equivalently, for $i \geq 2p$, the polynomial $f_{p,i}(n) = \beta_i(\mathsf{Cov}_{n,p}, \mathbb{F})$ satisfies $f_{p,i}(0) = 0$.

(f) *All roots of the polynomial $f_{p,i}$ are real and nonnegative (they are indeed integers). Moreover, the Euler characteristic of $\mathsf{Cov}_{n,p}$ is a polynomial in n with only real and positive roots.* □

Note that properties (a)–(d) are also true for $p = 4$ and $n \leq 7$; see Table 1. Moreover, by Proposition 10.2, property (a) is true whenever $p = n - 3$.

Proposition 11.1 suggests the following problem.

Question 11.2. *Among the six properties listed in Proposition* 11.1*, which of them hold for general p?*

We are particularly interested in knowing whether property (a) remains true for general p. First, this relates to the important problem of determining the connectivity degree of the complex of K_{n-p}-free graphs; this is the Alexander dual $\mathsf{Cov}^*_{n,p}$ of $\mathsf{Cov}_{n,p}$. Second, we would like to know more about connections between the complex Col^t_n of t-colorable graphs and $\mathsf{Cov}^*_{n,n-(t+1)}$; recall that the latter complex contains the former. As Linusson and Shareshian [14] observed, most homology of Col^t_n is concentrated in dimension $n(t-1) - \binom{t}{2} - 1 = d_{n,t+1}$ for all known examples, and so far no homology below this dimension has been found.

Regarding property (b), one may also ask whether the corresponding skeleton is VD or at least Cohen–Macaulay. Regarding property (c), we know that $\tilde{H}_{\binom{p+2}{2}-2}$ $(\mathsf{Cov}_{n,p}, \mathbb{Z})$ contains a free subgroup of rank $\binom{n-1}{p+1}$; use (7.1) and Corollary 7.6.

Since we do not have much data, it may well turn out that several of the properties in Proposition 11.1 do not generalize to larger values of p. We are particularly skeptical about properties (b) and (f).

For hypergraphs, the situation is even worse, as we have almost *no* data. Still, regarding property (a), one may ask whether it is true that $\mathsf{Cov}_{n,p,r}$ has no homology over any field strictly above dimension $\binom{p+r}{r} - 2$.

In our opinion, however, the most important open problem for $r \geq 3$ is to determine the maximum integer k for which $\Lambda_{k,p,r}$ has nonvanishing homology. This would give an upper bound on the degree of the polynomials $f_{p,r,i}$. Our hope is that the answer is pr, but we have no evidence whatsoever for this guess when $p, r \geq 3$. In particular, pr is not an upper bound on $\gamma(p,r)$; $\gamma(3,3) \geq 10$, as the hypergraph on the vertex set $\{0, 1, \dots, 9\}$ with edges $012, 234, 456, 678, 890$ is $(3,3)$-solid.

Complexity-theoretic remark. The $\mathsf{(VERTEX)\ COVER}$ problem of inputting a pair (G, p) is to determine whether $G \in \mathsf{Cov}_{n,p}$; n is the number of vertices in G. This is the *containment problem* for the family $\{\mathsf{Cov}_{n,p} : n, p \geq 1\}$. COVER is well-known to be NP-complete [6, 13]. A potentially interesting question is whether there is any deeper connection between this fact and the fact that the homology of $\mathsf{Cov}_{n,p}$ seems difficult to compute for general n and p.

Acknowledgments. I thank Anders Björner for many useful comments and suggestions and Volkmar Welker for fruitful discussions. I also thank an anonymous referee for several helpful remarks.

REFERENCES

[1] E. Babson, A. Björner, S. Linusson, J. Shareshian, and V. Welker, *Complexes of not i-connected graphs*, Topology, 38 (1999), pp. 271–299.

[2] C. Berge, *Graphs and Hypergraphs*, 2nd revised ed., North-Holland, Amsterdam, 1976.

[3] A. Björner, *Topological methods*, in Handbook of Combinatorics, R. Graham, M. Grötschel, and L. Lovász, eds., Elsevier, Amsterdam, 1995, pp. 1819–1872.

[4] B. Bollobás, *On generalized graphs*, Acta Math. Acad. Sci. Hungar., 16 (1965), pp. 447–452.

[5] M. K. Chari, *On discrete Morse functions and combinatorial decompositions*, Discrete Math., 217 (2000), pp. 101–113.

[6] S. Cook, *The complexity of theorem proving procedures*, in Proceedings of the Third Annual ACM Symposium on Theory of Computing, Shakur Heights, OH, 1971, pp. 151–158.

[7] J.-G. Dumas, F. Heckenbach, B. D. Saunders, and V. Welker, *Simplicial Homology, A Share Package for GAP*, manual, 2000.

[8] R. Forman, *Morse theory for cell complexes*, Adv. Math., 134 (1998), pp. 90–145.

[9] A. Hajnal, *A theorem on k-saturated graphs*, Canad. J. Math., 17 (1965), pp. 720–724.

[10] P. Hersh, *On optimizing discrete Morse functions*, Advances in Appl. Math., 35 (2005), pp. 294–322.

[11] J. Jonsson, *On the topology of simplicial complexes related to* 3*-connected and Hamiltonian graphs*, J. Combin. Theory Ser. A, 104 (2003), pp. 169–199.

[12] J. Jonsson, *Simplicial Complexes of Graphs*, Doctoral Thesis, KTH, Stockholm, Sweden, 2005.

[13] R. Karp, *Reducibility among combinatorial problems*, in Complexity of Computer Computations, R. Miller and J. Thatcher, eds., Plenum Press, New York, 1972, pp. 85–103.

[14] S. Linusson and J. Shareshian, *Complexes of t-colorable graphs*, SIAM J. Discrete Math., 16 (2003), pp. 371–389.

[15] S. Linusson, J. Shareshian, and V. Welker, *Complexes of graphs with bounded matching size*, submitted.

[16] L. Lovász, *Combinatorial Problems and Exercises*, 2nd ed., North-Holland, Amsterdam, 1993.

[17] J. S. Provan and L. J. Billera, *Decompositions of simplicial complexes related to diameters of convex polyhedra*, Math. Oper. Res., 5 (1980), pp. 576–594.

[18] J. Shareshian, *Links in the complex of separable graphs*, J. Combin. Theory Ser. A, 88 (1999), pp. 54–65.

[19] J. Shareshian, *Discrete Morse theory for complexes of* 2*-connected graphs*, Topology, 40 (2001), pp. 681–701.

[20] V. Turchin, *Homologies of complexes of doubly connected graphs*, Russian Math. Surveys (Uspekhi), 52 (1997), pp. 426–427.

[21] M. L. Wachs, *Topology of matching, chessboard, and general bounded degree graph complexes*, Algebra Universalis, 49 (2003), pp. 345–385.

SIAM J. Discrete Math.
Vol. 19, No. 3, pp. 651–663

DECYCLING CARTESIAN PRODUCTS OF TWO CYCLES[*]

DAVID A. PIKE[†] AND YUBO ZOU[†]

Abstract. The decycling number $\nabla(G)$ of a graph G is the smallest number of vertices which can be removed from G so that the resultant graph contains no cycles. In this paper, we study the decycling number for the family of graphs consisting of the Cartesian product of two cycles. We completely solve the problem of determining the decycling number of $C_m \Box C_n$ for all m and n. Moreover, we find a vertex set T that yields a maximum induced tree in $C_m \Box C_n$.

Key words. decycling, cycle, Cartesian product, maximum induced forest

AMS subject classifications. 05C38, 05C05, 94C15

DOI. 10.1137/S089548010444016X

1. Introduction.

In 1986, Erdős, Saks, and Sós published a paper in which they considered the problem of finding, for a given graph G, the size $t(G)$ of a maximum subset T of $V(G)$ that would induce a tree [8]. Meanwhile, the more general problem of finding the size of a maximum subset F of $V(G)$ that would induce a forest was also beginning to receive attention for various types of graphs, such as cubic graphs [6, 10] and planar graphs [1, 2].

The problem of finding the size of a maximum subset F of $V(G)$ that induces a forest can be reformulated as the problem of determining the size of a minimum subset S of $V(G)$ for which $G - S$ is acyclic. Any set S for which $G - S$ contains no cycles is a *decycling set*. The size of a minimum decycling set S in a graph G is the *decycling number* of G and will be denoted by $\nabla(G)$. Note that decycling sets are sometimes also called *feedback vertex sets*, and they have applications in areas such as circuit design and deadlock prevention (see [9]).

In [5], various introductory results were presented, followed by investigations into hypercubes as well as 2-dimensional grid graphs. The 2-dimensional grid graph $P_m \Box P_n$ is the Cartesian product of a path P_m on m vertices and a path P_n on n vertices (here we follow the notation presented in [11] for Cartesian products). Further results concerning $\nabla(P_m \Box P_n)$ were subsequently presented in [4] and summarized in a survey paper on decycling [3].

In [4] and again in [3], determining the decycling number for the Cartesian product of two cycles, i.e., $\nabla(C_m \Box C_n)$, is presented as an open problem. In this paper we completely solve this problem, determining $\nabla(C_m \Box C_n)$ for all $m \geqslant 3$ and $n \geqslant 3$. Further, for each combination of m and n other than $m = n = 4$, we show how to construct a minimum decycling set S of size $\nabla(C_m \Box C_n)$, such that $T = V(C_m \Box C_n) - S$ is the vertex set of a maximum induced tree in $C_m \Box C_n$.

Before moving on to our results, we review some of the introductory results that appeared first in [5].

Lemma 1.1. *If G is a connected graph with p vertices ($p > 2$), q edges, and*

[*] Received by the editors January 22, 2004; accepted for publication (in revised form) March 30, 2005; published electronically November 15, 2005.
http://www.siam.org/journals/sidma/19-3/44016.html

[†] Department of Mathematics and Statistics, Memorial University of Newfoundland, St. John's, NL, Canada A1C 5S7 (dapike@math.mun.ca, yubo@math.mun.ca). The work of the first author was supported by NSERC.

maximum degree Δ, *then*

$$\nabla(G) \geqslant \frac{q-p+1}{\Delta-1}\,.$$

THEOREM 1.2. *If* G *and* H *are homeomorphic graphs, then* $\nabla(G) = \nabla(H)$.

2. Decycling of $C_m \Box C_n$ (initial cases). In the next two sections, we investigate the decycling number of the graph $C_m \Box C_n$, the Cartesian product of a cycle C_m on m vertices and a cycle C_n on n vertices. In this section we establish lower bounds on $\nabla(C_m \Box C_n)$ and obtain the exact results for several initial cases. In the next section, we obtain the general result of the decycling set of the graph $C_m \Box C_n$, for all m and n.

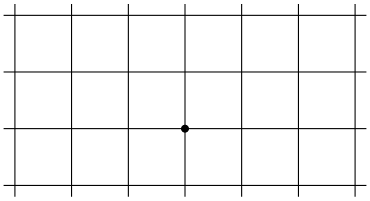


FIG. 2.1. *$C_4 \Box C_7$, Cartesian product of C_4 and C_7.*

Following the notation in [5], it will be useful to have a standard labeling for the vertices of $C_m \Box C_n$, and we choose one that corresponds to matrix notation: the ith vertex in the jth copy of C_m will be denoted $v_{i,j}$. Figure 2.1 is a simple example of $C_4 \Box C_7$, and the vertex labelled by "•" is denoted by $v_{3,4}$. Note that $C_m \Box C_n$ is a 4-regular graph, and hence by Lemma 1.1, we have the following lower bound for the size of any decycling set of $C_m \Box C_n$.

LEMMA 2.1. *For the graph* $C_m \Box C_n$,

$$\nabla(C_m \Box C_n) \geqslant \frac{mn+1}{3}. \tag{2.1}$$

Carrying the matrix analogy further, we sometimes speak of the copies of C_m and C_n as the columns and rows, respectively, of $C_m \Box C_n$. In order that decycling will be more readily recognizable in our figures, we frequently emphasize only the vertices of the decycling set.

Because the result in $C_4 \Box C_n$ is different from other $C_m \Box C_n$, we dispose of this case prior to developing more general results.

THEOREM 2.2. $\nabla(C_4 \Box C_n) = \lceil \frac{3n}{2} \rceil$ *for all* $n \geqslant 3$.

Proof. Every column of the graph is a 4-cycle, so we must remove at least one vertex in each column for decycling. For any two adjacent columns in $C_4 \Box C_n$, we need to remove at least 3 vertices to decycle these two columns, and so if n is even, then $\nabla(C_4 \Box C_n) \geqslant \frac{3n}{2}$. When n is even, we can find a decycling set of that size. Let $M = \bigcup_{i=1}^{k-1}(\{v_{2,4i-3},\ v_{4,4i-3},\ v_{3,4i-2},\ v_{1,4i-1},\ v_{3,4i-1},\ v_{2,4i}\})$, where $k = \lceil \frac{n}{4} \rceil$. Then $S = M \cup \{v_{2,n-1},\ v_{4,n-1},\ v_{1,n}\}$ is a decycling set for $C_4 \Box C_n$, where $n \equiv 2 \pmod 4$ (see Figure 2.2 for the case $n = 6$). If $n \equiv 0 \pmod 4$ and $n > 4$, then $S = M \cup \{v_{2,n-3},\ v_{4,n-3},\ v_{1,n-2},\ v_{2,n-1},\ v_{3,n-1},\ v_{1,n}\}$ is a minimum decycling set (see Figure 2.3 for the case $n = 8$).

Observe that in each case the vertices of $V(C_m \square C_n) - S$ induce a tree. However, for $C_4 \square C_4$, no minimum decycling set will yield a forest with only one component. In this case, a maximum induced tree has 9 vertices and is produced by the nonminimum decycling set $\{v_{3,1},\ v_{4,1},\ v_{2,2},\ v_{1,3},\ v_{3,3},\ v_{1,4},\ v_{4,4}\}$. A maximum induced forest having 10 vertices is obtained with the decycling set $\{v_{1,1},\ v_{3,1},\ v_{2,2},\ v_{1,3},\ v_{3,3}, v_{4,4}\}$.

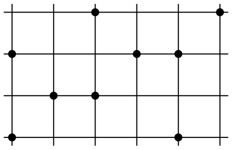


FIG. 2.2. *Decycling set of* $C_4 \square C_6$.

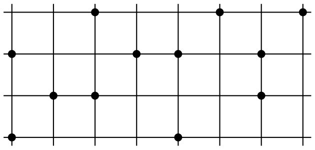


FIG. 2.3. *Decycling set of* $C_4 \square C_8$.

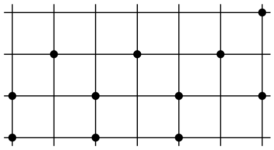


FIG. 2.4. *Decycling set of* $C_4 \square C_7$.

If n is odd, in every two adjacent columns, we still must remove at least 3 vertices. If we attempt to remove exactly 3 vertices from each pair of columns $(i, i+1)$ for $1 \leqslant i < n-1$, then we find that removing only 1 vertex from column n leaves a cycle in the graph (one of the pairs $(n-1, n)$ or $(n, 1)$ will contain only 2 vertices of the decycling set). It follows that at least $3(\frac{n-1}{2})+2 = \lceil \frac{3n}{2} \rceil$ vertices will have to be removed in order to decycle $C_4 \square C_n$ where n is odd. A decycling set of this size, and whose complement induces a tree, exists, namely $\bigcup_{i=1}^{k}(\{v_{3,2i-1},\ v_{4,2i-1},\ v_{2,2i}\}) \cup \{v_{1,n},\ v_{3,n}\}$, where $k = \frac{n-1}{2}$. (See Figure 2.4 for the case $n = 7$.) □

Throughout the remainder of this section, we assume that $4 \notin \{m, n\}$.

In [5], the *outlay* of a set S of vertices in a graph G is defined as

$$\theta(S) := \sigma(S) - |S| - \epsilon(S) - \omega(G - S) + 1\,,$$

where $\sigma(S)$ is the sum of the degrees of the vertices in S, $\epsilon(S)$ is the number of edges in the induced subgraph $G[S]$, and $\omega(G - S)$ is the number of components in $G - S$.

By Lemma 1.3 in [5], if G is a connected graph with p vertices and q edges, and S is any decycling set for G, then $\theta(S) = q - p + 1$. For $C_m \square C_n$, we can easily compute $mn + 1 = \theta(S) = 3|S| + 1 - (\epsilon + \omega)$, so

$$3|S| = mn + (\epsilon + \omega)\,. \tag{2.2}$$

Since $\omega \geqslant 1$, therefore $|S| \geqslant \frac{mn+1}{3}$, which in turn yields (2.1). We will now show that $\epsilon + \omega > 1$, thereby obtaining a greater lower bound on $\nabla(C_m \square C_n)$ than is given by Lemma 2.1.

LEMMA 2.3. *For $G = C_m \square C_n$, if S is a minimum decycling set, then*

$$\epsilon(S) + \omega(G - S) \geqslant 2\,.$$

Proof. By the symmetry of the graph $C_m \square C_n$, we can consider the following 6 cases only:

(1) $m \equiv 0 \pmod 3$, $n \equiv 0 \pmod 3$;
(2) $m \equiv 1 \pmod 3$, $n \equiv 0 \pmod 3$;
(3) $m \equiv 1 \pmod 3$, $n \equiv 1 \pmod 3$;
(4) $m \equiv 2 \pmod 3$, $n \equiv 0 \pmod 3$;
(5) $m \equiv 2 \pmod 3$, $n \equiv 1 \pmod 3$;
(6) $m \equiv 2 \pmod 3$, $n \equiv 2 \pmod 3$.

By considering (2.2), modulo 3, we quickly find that $\epsilon + \omega \geqslant 2$ for cases (3) and (6), and $\epsilon + \omega \geqslant 3$ for cases (1), (2), and (4). We therefore now consider only case (5).

For each 4-cycle (a, b, c, d) in $C_m \Box C_n$, we now create the block $\{a, b, c, d\}$, and we let $\mathcal{B}$ be the set of all such blocks. Let $\mathcal{H}$ be the bipartite graph with bipartition $(V, \mathcal{B})$, where $V = V(C_m \Box C_n)$ and in which $v \in V$ is adjacent to $B \in \mathcal{B}$ if and only if $v \in B$.

Given a subset S of V, we also consider the related bipartite graph $\mathcal{H} - S$ having bipartition $(V - S, \mathcal{B} - S)$, where $\mathcal{B} - S = \{B - S : B \in \mathcal{B}\}$ and in which $v \in V - S$ is adjacent to $B \in \mathcal{B} - S$ if and only if $v \in B$. Whenever S is a decycling set of $C_m \Box C_n$, each vertex $v \in V - S$ will have degree 4 in $\mathcal{H} - S$, yet the mn vertices of $\mathcal{B} - S$ must each have degree at most 3 (as blocks, the mn elements of $\mathcal{B} - S$ must each have size at most 3).

Assuming that $\epsilon + \omega = 1$, then $\epsilon = 0$, $\omega = 1$, and so (2.2) yields

$$\nabla(C_m \Box C_n) = \frac{mn + 1}{3}. \tag{2.3}$$

Since $\varepsilon = 0$, it follows that each of the mn blocks of $\mathcal{B} - S$ must have size of 2 or 3. Let us assume that x of the vertices of $\mathcal{B} - S$ have degree 2 in $\mathcal{H} - S$, so the remaining $mn - x$ vertices of $\mathcal{B} - S$ each have degree 3 in $\mathcal{H} - S$. Counting the number of edges in $\mathcal{H} - S$, we have

$$\begin{aligned} 4(mn - \nabla) &= 2x + 3(mn - x) \\ \Rightarrow x &= \frac{mn + 4}{3}. \end{aligned}$$

At this point, each vertex of $(C_m \Box C_n) - S$ must have degree 1, 2, 3, or 4 (a vertex of degree 0 is impossible because $\omega = 1$). Those of degree 2 can be further classified, depending on whether the 2 neighbors are in orthogonal directions or in the same lateral direction. Thus we can now identify each vertex of $(C_m \Box C_n) - S$ as being of type 1, 2o, 2l, 3, or 4, as illustrated in Figure 2.5. (Since $\epsilon = 0$, there exist only these 5 types of vertices in $(C_m \Box C_n) - S$.) Throughout this lemma, in order that the vertices in the decycling set will be more readily differentiable from the vertices of $(C_m \Box C_n) - S$ in the figure, we use "•" to denote the vertices in $C_m \Box C_n$, and a vertex surrounded by "□" indicates that the vertex is in the decycling set.

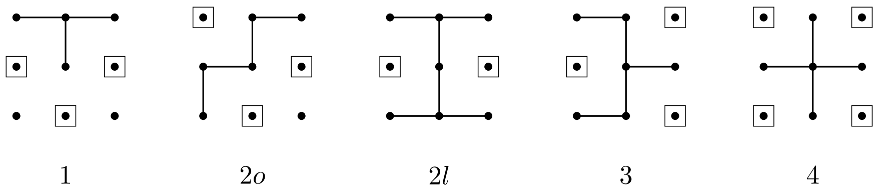


FIG. 2.5. *The 5 types of vertices in $G - S$ (the vertex in the center is the vertex that we mean for the type, and the vertices surrounded by "□" denote vertices which are in the decycling set).*

Every type-1 vertex of $G - S$ is in two blocks of size 2 in $\mathcal{B} - S$ and two blocks of size 3, every type-$2o$ vertex is in one block of size 2 and three blocks of size 3, and each remaining vertex of $(C_m \Box C_n) - S$ is in four blocks of size 3. Suppose that we have x_1, x_{2o}, x_{2l}, x_3, x_4 vertices of type-1, $2o$, $2l$, 3, 4, respectively. Counting the number of edges in $\mathcal{H} - S$ which are incident to the blocks of size 2 in $\mathcal{B} - S$, we have

$$2x_1 + x_{2o} = 2x = \frac{2mn+8}{3}. \tag{2.4}$$

Similarly, counting the number of edges which are incident to the blocks of size 3, we have

$$\begin{aligned} 2x_1 + 3x_{2o} + 4x_{2l} + 4x_3 + 4x_4 &= 3(mn - x) \\ &= 3mn - 3\left(\frac{mn+4}{3}\right) \\ &= 2mn - 4. \end{aligned} \tag{2.5}$$

Counting the number of edges of $C_m \Box C_n$ with one endpoint in S and the other in $V - S$, we have

$$3x_1 + 2x_{2o} + 2x_{2l} + 1x_3 + 0x_4 = 4\nabla. \tag{2.6}$$

Solving the system of equations consisting of (2.4)–(2.6), we find that

$$x_4 = 1 + x_{2l}.$$

Since $x_{2l} \geqslant 0$, then there is at least one type-4 vertex. Note that each type-4 vertex can only be adjacent in $G - S$ to vertices that are type-1 or type-$2l$. Since $\omega(G - S) = 1$, each type-4 vertex must have at least one type-$2l$ neighbor. Moreover, since $x_4 = 1 + x_{2l}$, it follows that the type-$2l$ and type-4 vertices induce a tree in which the vertices are alternately type-4 and type-$2l$. The neighbors of this tree in $G - S$ are all type-1. But $\omega(G - S) = 1$, and hence we now deduce that $x_{2o} = x_3 = 0$, yielding a solution to (2.4)–(2.6):

$$\begin{cases} x_1 &= \dfrac{mn+4}{3}, \\ x_{2l} &= \dfrac{mn-8}{6}, \\ x_4 &= \dfrac{mn-2}{6}. \end{cases}$$

Consider the vertices of an arbitrary row in G. Since $\varepsilon = 0$, each vertex of S in the row will be incident with 2 horizontal edges of G that do not appear in $G - S$. The horizontal edges that do appear in $G - S$ form a collection of horizontal paths, each of which has two endpoints that are both type-1 vertices, whereas the interior vertices are alternately type-4 and type-$2l$. Each horizontal path thus contains an even number of edges. Of the n horizontal edges that appeared in this row in G, an even number do not appear in $G - S$ while an even number do. Hence n must be even, or else we have a contradiction. By considering an arbitrary column of G, we similarly find that m must be even.

Now we consider the vertices in the decycling set. At this point, the vertices in S must have 0, 1, or 2 type-$2l$ neighbors. Those having no type-$2l$ neighbor can be

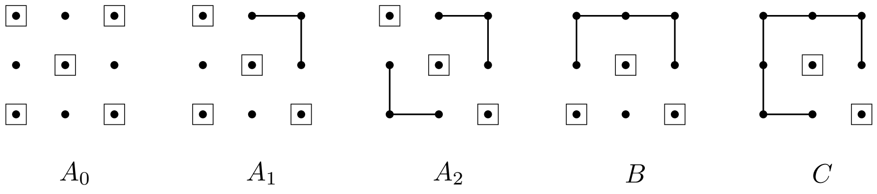


Fig. 2.6. *The* 5 *types of vertices in* S *(the vertex in the center is the vertex that we mean for the type, and the vertices surrounded by "□" denote vertices which are in the decycling set).*

further classified, depending on the number of type-4 diagonal neighbors, where we define a *diagonal neighbor* of a vertex u to be any vertex that is at distance 2 from u in a 4-cycle of G. The resultant 5 possible types of vertices in S are illustrated in Figure 2.6.

For each type-A_k vertex ($k = 0, 1, 2$), its 4 neighbors are all type-1 vertices. Each type-B vertex has one type-$2l$ neighbor and the other three are type-1. For each type-C vertex, two of its neighbors (in orthogonal directions) are type-$2l$ and the other two are type-1. Since both m and n are even, then G has an even number of rows and even number of columns, so we label the rows and columns of G such that all of the type-4 vertices are in rows and columns with even parity. Consequently, all type-A_0 vertices are in rows and columns with even parity while all vertices of type-A_1, A_2, B, or C are in rows and columns with odd parity. Type-1 and type-$2l$ vertices have row and column labels with different parities. Hence all of the vertices of S that are diagonal neighbors of any vertex of type-A_1, A_2, B, or C must be of type-A_0.

Note that all $\frac{mn}{4}$ of the vertices in rows and columns, each having odd parity (i.e., the vertices of type-A_1, A_2, B, and C), are in the decycling set. And since $\varepsilon = 0$, all $\frac{mn}{2}$ of the vertices in rows and columns having different parities (i.e., the type-1 and type-$2l$ vertices) are not in S. Thus, if we let T denote the set of all vertices that are of type-A_1, A_2, B, or C, then $(C_m \square C_n) - T$ is homeomorphic to $C_{\frac{m}{2}} \square C_{\frac{n}{2}}$, and so by Theorem 1.2,

$$\begin{aligned} \nabla(C_m \square C_n) &= \nabla\big((C_m \square C_n) - T\big) + |T| \\ &= \nabla(C_{\frac{m}{2}} \square C_{\frac{n}{2}}) + \frac{mn}{4}. \end{aligned} \tag{2.7}$$

More generally, we find that

$$\nabla(C_{m_i} \square C_{n_i}) = \nabla(C_{m_{i-1}} \square C_{n_{i-1}}) + m_{i-1}n_{i-1},$$

where $m_i = \frac{m}{2^{k-i}}$ and $n_i = \frac{n}{2^{k-i}}$ for $i = 0, 1, \ldots, k$, and k is the least nonnegative integer such that at least one of $\frac{m}{2^k}$ or $\frac{n}{2^k}$ is either odd or equal to 4. Further, observe that $\nabla(C_{m_i} \square C_{n_i}) = \frac{m_i n_i + 1}{3}$ if and only if $\nabla(C_{m_{i-1}} \square C_{n_{i-1}}) = \frac{m_{i-1} n_{i-1} + 1}{3}$. But $\nabla(C_{m_0} \square C_{n_0}) \neq \frac{m_0 n_0 + 1}{3}$, since either one of m_0 or n_0 is odd or else Theorem 2.2 applies. We therefore have a contradiction to (2.3) and hence $\varepsilon(S) + \omega(G - S) \geqslant 2$ for case (5). By again considering (2.2), modulo 3, we now conclude that $\varepsilon(S) + \omega(G - S) \geqslant 4$. □

Theorem 2.4. $\nabla(C_m \square C_n) \geqslant \lceil \frac{mn+2}{3} \rceil$.

Proof. By Lemma 2.3 and (2.2), we can easily get the desired result. □

The reduction technique described towards the end of the proof of Lemma 2.3 can also be employed as a doubling construction, which we now present.

Theorem 2.5. *Suppose S is any minimum decycling set of $C_m\Box C_n$ with $\nabla(C_m\Box C_n)=\lceil\frac{mn+2}{3}\rceil$. Let $S'=\{v_{2i-1,2j-1} : v_{i,j}\in S\}$ be a vertex set in $C_{2m}\Box C_{2n}$, and $T=\{v_{i,j} : i=2,4,\dots,2m;\ j=2,4,\dots,2n\}$. Then $S'\cup T$ is a minimum decycling set of $C_{2m}\Box C_{2n}$. Furthermore, if $\omega\big((C_m\Box C_n)-S\big)=1$, then there exists a minimum decycling set S'' of $C_{2m}\Box C_{2n}$ such that $\omega\big((C_{2m}\Box C_{2n})-S''\big)=1$.*

Proof. Note that $(C_{2m}\Box C_{2n})-T$ is a graph homeomorphic to $C_m\Box C_n$. Hence, $\big((C_{2m}\Box C_{2n})-T\big)-S'=(C_{2m}\Box C_{2n})-(S'\cup T)$ is acyclic. Therefore, $S'\cup T$ is a decycling set of $C_{2m}\Box C_{2n}$.

Since, by Theorem 2.4,

$$\begin{aligned}\nabla(C_{2m}\Box C_{2n}) &\geqslant \left\lceil\frac{4mn+2}{3}\right\rceil\\ &= mn+\left\lceil\frac{mn+2}{3}\right\rceil\\ &= |T|+|S'|,\end{aligned}$$

$S'\cup T$ is a minimum decycling set of $C_{2m}\Box C_{2n}$.

Clearly $\varepsilon(S'\cup T)=0$. In order to obtain a maximum induced tree in $C_{2m}\Box C_{2n}$, we therefore must transform the decycling set $S'\cup T$ into a new decycling set S'' such that $\varepsilon(S'')=\varepsilon(S)$.

Suppose that for some i and j, both of $v_{i,j}$ and $v_{i,j+1}$ are in S. Then $v_{2i-1,2j}$ is an isolated vertex in $(C_{2m}\Box C_{2n})-(S'\cup T)$. Also, $v_{2i,2j}\in T$ and $v_{2i+1,2j}\notin(S'\cup T)$. By removing $v_{2i,2j}$ from the decycling set and replacing it with $v_{2i+1,2j}$, we obtain a new minimum decycling set. If $\varepsilon(S)\geqslant 1$, then by performing a suitable combination of at most $\varepsilon(S)$ similar transformations, we obtain a minimum decycling set S'' for which $\varepsilon(S'')=\varepsilon(S)$ and $\omega\big((C_{2m}\Box C_{2n})-S''\big)=1$. □

We now carry on with several initial cases.

Lemma 2.6. $\nabla(C_3\Box C_n)=\lceil\frac{3n+2}{3}\rceil$.

Proof. By Theorem 2.4, we get $\nabla(C_3\Box C_n)\geqslant\lceil\frac{3n+2}{3}\rceil=n+1$. Let $k=\lfloor\frac{n}{3}\rfloor$, $M=\bigcup_{i=1}^{k}\{v_{1,3i-2},\ v_{2,3i-1},\ v_{3,3i}\}$, $S_0=\{v_{2,n}\}$, $S_1=\{v_{2,n},\ v_{3,n}\}$, and $S_2=\{v_{1,n-1},\ v_{2,n-1},\ v_{2,n}\}$. Then $M\cup S_t$ is a decycling set of size $n+1$ for $C_3\Box C_n$ with the property that $\omega\big((C_3\Box C_n)-(M\cup S_t)\big)=1$, where $n\equiv t \pmod 3$. □

Lemma 2.7. $\nabla(C_8\Box C_n)=\lceil\frac{8n+2}{3}\rceil$.

Proof. By Theorem 2.4, we have the lower bound $\nabla(C_8\Box C_n)\geqslant\lceil\frac{8n+2}{3}\rceil$. We should divide the number of columns into 6 cases modulo 6. Let $k=\lfloor\frac{n}{3}\rfloor$, and let $M=\bigcup_{i=1}^{k}\{v_{1,3i-2},\ v_{4,3i-2},\ v_{7,3i-2},\ v_{2,3i-1},\ v_{5,3i-1},\ v_{3,3i},\ v_{6,3i},\ v_{8,3i}\}$. For $n\equiv 3 \pmod 6$, $M\cup\{v_{1,n}\}$ is a minimum decycling set. For $n\equiv 1 \pmod 6$, $M\cup\{v_{2,n},\ v_{5,n},\ v_{6,n},\ v_{8,n}\}$ is a minimum decycling set. For $n\equiv 4 \pmod 6$, $M\cup\{v_{1,n},\ v_{3,n},\ v_{5,n},\ v_{7,n}\}$ is a minimum decycling set. For $n\equiv 5 \pmod 6$, $M\cup\{v_{2,n-1},\ v_{5,n-1},\ v_{7,n-1},\ v_{3,n},\ v_{5,n},\ v_{8,n}\}$ is a minimum decycling set. For $n\equiv 0 \pmod 6$, $\big(M-\{v_{2,n-1}\}\big)\cup\{v_{3,n-1},\ v_{2,n}\}$ is a minimum decycling set. For $n\equiv 2 \pmod 6$, $\bigcup_{i=1}^{k-1}\{v_{1,3i-2},\ v_{4,3i-2},\ v_{7,3i-2},\ v_{2,3i-1},\ v_{5,3i-1},\ v_{3,3i},\ v_{6,3i},\ v_{8,3i}\}\cup\{v_{1,n-4},\ v_{4,n-4},\ v_{7,n-4},\ v_{2,n-3},\ v_{5,n-3},\ v_{1,n-2},\ v_{4,n-2},\ v_{7,n-2},\ v_{2,n-1},\ v_{6,n-1},\ v_{8,n-1},\ v_{3,n},\ v_{5,n},\ v_{8,n}\}$ is a minimum decycling set. Note that in each of these six cases, the corresponding maximum induced forest consists of a single tree. □

Lemma 2.8. *If $m\equiv 0 \pmod 3$, then $\nabla(C_m\Box C_n)=rn+1$, where $m=3r$.*

Proof. By Theorem 2.4, we have the lower bound, $\nabla(C_{3r}\Box C_n)\geqslant\lceil\frac{3rn+2}{3}\rceil=rn+1$. Now if n is odd, then let $M=\bigcup_{i=1}^{r}\big(\{v_{3i-2,1}\}\cup\bigcup_{j=1}^{k}\{v_{3i-1,2j},\ v_{3i,2j+1}\}\big)$, where $n=2k+1$. Considering the graph $(C_{3r}\Box C_n)-M$, rows $3i-2$, $3i-1$, and $3i$

have a path from $v_{3i-2,2}$ to $v_{3i,2}$ for each $1 \leqslant i \leqslant r$. By joining these paths, we have a cycle C of length $r(n+3)$ starting and ending at $v_{1,2}$. Each vertex not on cycle C and not in M has one neighbor in C and three neighbors in M. So $(C_{3r} \Box C_n) - M$ consists of the cycle C with $r(n-3)$ pendant edges and $6r$ vertices of degree 2. (See Figure 2.7 for an example of the left-over graph of $(C_6 \Box C_{11}) - M$.)

If n is even and r is odd, let $M = \bigcup_{i=1}^{r} (\{v_{3i-2,1},\ v_{3i-2,n-2},\ v_{3i-1,n},\ v_{3i,n-1}\} \cup \bigcup_{j=1}^{k} \{v_{3i-1,2j+1},\ v_{3i,2j}\})$, where $n = 2k+4$. Using a similar method to the above case, $(C_{3r} \Box C_n) - M$ consists of a cycle C of length $r(n+6)$ with $r(n-6)$ pendant edges and $12r$ vertices of degree 2. (See Figure 2.8 for an example of the left-over graph of $(C_9 \Box C_{14}) - M$.)

In either of the above cases, $|M| = rn$ and $(C_m \Box C_n) - M$ is connected and contains only one cycle. Now by letting S consist of M plus any degree-2 vertex of the cycle C of $(C_m \Box C_n) - M$, we obtain a minimum decycling set, the complement of which induces a tree.

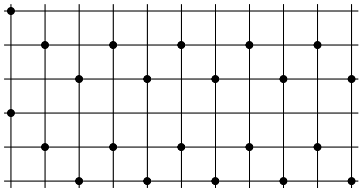


FIG. 2.7. *The left-over graph (with one cycle) of $(C_{3r} \Box C_n) - M$, where n is odd.*

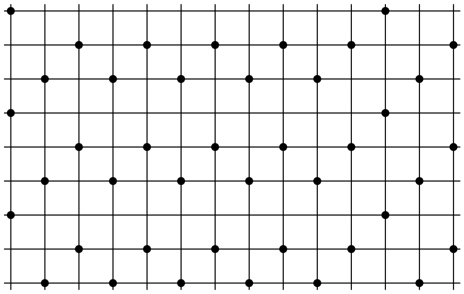


FIG. 2.8. *The left-over graph (with one cycle) of $(C_{3r} \Box C_n) - M$, where n is even and r is odd.*

Now suppose that r and n are both even, and let k be the least nonnegative integer such that at least one of $\frac{r}{2^k}$ or $\frac{n}{2^k}$ is odd or $\frac{n}{2^k}$ equals 8, and let $m_i = \frac{m}{2^{k-i}}$ and $n_i = \frac{n}{2^{k-i}}$ for each $i = 0, 1, \ldots, k$. Then we can find a minimum decycling set S_0 of cardinality $\frac{rn}{2^{2k}} + 1$ in $C_{m_0} \Box C_{n_0}$ such that $\omega\big((C_{m_0} \Box C_{n_0}) - S_0\big) = 1$. Now, for each $i = 0,\ 1,\ \ldots,\ k-1$, apply Theorem 2.5 to $C_{m_i} \Box C_{n_i}$ to construct a minimum decycling set S_{i+1} of size $\nabla(C_{m_0} \Box C_{n_0}) + \frac{mn}{2^{2k}} \sum_{j=0}^{i} 4^j$ in $C_{m_{i+1}} \Box C_{n_{i+1}}$ such that $\omega\big((C_{m_{i+1}} \Box C_{n_{i+1}}) - S_{i+1}\big) = 1$. It follows that $\nabla(C_m \Box C_n) = \nabla(C_{m_0} \Box C_{n_0}) + \frac{mn}{2^{2k}} \sum_{j=0}^{k-1} 4^j = rn + 1$. □

By the symmetry of the graph $C_m \Box C_n$, in the rest of this paper, we do not need to consider the case of $n \equiv 0 \pmod 3$ for any choice of m.

LEMMA 2.9. $\nabla(C_5 \Box C_n) = \left\lceil \frac{5n+2}{3} \right\rceil$.

Proof. By Theorem 2.4, we have $\nabla(C_5 \Box C_n) \geqslant \lceil \frac{5n+2}{3} \rceil$. Let $k = \lfloor \frac{n}{3} \rfloor$, $M = \bigcup_{i=1}^{k} \{v_{1,3i-2},\ v_{3,3i-2},\ v_{2,3i-1},\ v_{4,3i-1},\ v_{5,3i}\}$, $S_1 = \{v_{1,n},\ v_{3,n},\ v_{4,n}\}$, and $S_2 = \{v_{1,n-1},\ v_{3,n-1},\ v_{4,n},\ v_{5,n}\}$. We can easily verify that $M \cup S_t$ is a decycling set of size $\lceil \frac{5n+2}{3} \rceil$ for $C_5 \Box C_n$ such that $\omega\big((C_5 \Box C_n) - (M \cup S_t)\big) = 1$, where $n \equiv t \pmod 3$. □

LEMMA 2.10. $\nabla(C_7 \Box C_n) = \lceil \frac{7n+2}{3} \rceil$.

Proof. By Theorem 2.4, we have $\nabla(C_7 \Box C_n) \geqslant \lceil \frac{7n+2}{3} \rceil$. Let $k = \lfloor \frac{n}{3} \rfloor$. For $n \equiv 1 \pmod 3$, let $S = \bigcup_{i=1}^{k-1} \{v_{2,3i-2},\ v_{6,3i-2},\ v_{3,3i-1},\ v_{5,3i-1},\ v_{7,3i-1},\ v_{1,3i},\ v_{4,3i}\} \cup \{v_{3,n-3},\ v_{5,n-3},\ v_{7,n-3},\ v_{2,n-2},\ v_{5,n-2},\ v_{1,n-1},\ v_{3,n-1},\ v_{6,n-1},\ v_{4,n},\ v_{7,n}\}$. For $n \equiv 2 \pmod 3$, let $S = \bigcup_{i=1}^{k} \{v_{1,3i-2},\ v_{5,3i-2},\ v_{2,3i-1},\ v_{4,3i-1},\ v_{7,3i-1},\ v_{3,3i},\ v_{6,3i}\} \cup \{v_{1,n-1},\ v_{4,n-1},\ v_{7,n-1},\ v_{1,n},\ v_{3,n},\ v_{6,n}\}$. In both cases, S is a decycling set of size

$\lceil \frac{7n+2}{3} \rceil$ and $\omega((C_7 \Box C_n) - S) = 1$. □

According to Lemmas 2.6–2.10, we have the following theorem.

THEOREM 2.11. *Let m and n be integers such that $m \in \{5, 7, 8\} \cup \{3, 6, 9, \dots\}$, $n \geqslant 3$, and $n \neq 4$. Then $\nabla(C_m \Box C_n) = \lceil \frac{mn+2}{3} \rceil$, and $t(C_m \Box C_n) = mn - \nabla(C_m \Box C_n)$.*

3. Decycling $C_m \Box C_n$ (the general cases). In the previous section, we discussed the decycling set of $C_m \Box C_n$ for small values of m. In this section, we investigate all of the remaining cases. Since we already solved the problem for $m \equiv 0 \pmod 3$, we only need to consider four cases (modulo 6), i.e., $m \equiv 1 \pmod 6$, $m \equiv 2 \pmod 6$, $m \equiv 4 \pmod 6$, $m \equiv 5 \pmod 6$, and by the symmetry of $C_m \Box C_n$, for each case, we do not need to consider the case of $n \equiv 0 \pmod 3$. Unless stated otherwise, throughout this section we assume that $4 \notin \{m, n\}$.

LEMMA 3.1. *If $m = 6r + 1$, then $\nabla(C_m \Box C_n) = 2rn + \lceil \frac{n+2}{3} \rceil$.*

Proof. If $r = 1$, then the result follows from Lemma 2.10. For $r > 1$, we employ an iterative construction in which we add 6 new rows at a time, starting with the graph $C_7 \Box C_n$. This iterative technique takes advantage of a particular configuration of three consecutive rows; this configuration is initially present in the decycling set described in Lemma 2.10 for $C_7 \Box C_n$, and a copy of the configuration is produced with each iteration. After $(r-1)$ iterations, we will have constructed a decycling set of size $2rn + \lceil \frac{n+2}{3} \rceil$ in $C_m \Box C_n$. From Theorem 2.4, we have $\nabla(C_m \Box C_n) \geqslant \lceil \frac{mn+2}{3} \rceil = \lceil \frac{(6r+1)n+2}{3} \rceil = 2rn + \lceil \frac{n+2}{3} \rceil$, which tells us that the decycling set we construct is optimal.

We now consider two subcases, each of which employs its own configuration of three consecutive rows from Lemma 2.10.

(i) $n \equiv 1 \pmod 3$.

Beginning with the decycling set of $C_7 \Box C_n$, we say that a row is type-α if its deleted vertices are in the same columns as those of the fifth row of $C_7 \Box C_n$ which was described in Lemma 2.10 (i.e., row j is type-α if the set of vertices removed from it is $\bigcup_{i=1}^{t} \{v_{j,3i-1}\} \cup \{v_{j,n-3}\}$, where $n = 3t + 1$). Similarly, type-β (resp., type-γ) rows are those with a configuration identical to that of the sixth $(\bigcup_{i=1}^{t-1} \{v_{j,3i-2}\} \cup \{v_{j,n-1}\})$ (resp., seventh $(\bigcup_{i=1}^{t-1} \{v_{j,3i-1}\} \cup \{v_{j,n-3}, v_{j,n}\})$) row of $C_7 \Box C_n$.

Focusing on the three consecutive rows that are, in order, of type-α, β, γ in $C_{7+6k} \Box C_n$, for some $k \geqslant 0$, we now describe how to insert six new rows and obtain a minimum decycling set of $C_{7+6(k+1)} \Box C_n$. Following the row of type-α in $C_{7+6k} \Box C_n$, insert three new rows, the first two being of type-β and type-γ, respectively. For the third of these new rows, add the vertices in columns $3i$ (where $i = 1, 2, \dots, t-1$) and $n-2$ to the decycling set.

Now, following the original type-β row, insert another three new rows. For the first of these three new rows, we select the vertices in columns $3i$ (where $i = 1, 2, \dots, t-1$), $n-2$, and n to add to the decycling set. For the second row, select the vertices in columns $3i-1$ (where $i = 1, 2, \dots, t-1$) and $n-3$. For the third row, delete vertices so that it is a type-β row (see Figure 3.1 for the expansion pattern where $n = 13$). We now have a minimum decycling set of $C_{7+6(k+1)} \Box C_n$.

Note that the new graph, $C_{7+6(k+1)} \Box C_n$, contains three consecutive rows that are of type-α, β, γ, and hence we can iterate this expansion procedure.

(ii) $n \equiv 2 \pmod 3$.

Similar to the previous case, start from the decycling set of $C_7 \Box C_n$. A row is type-α if its deleted vertices are in the same columns as those of the fourth

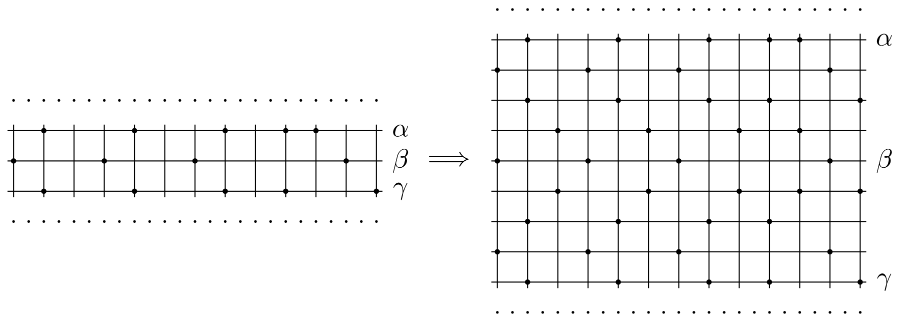


FIG. 3.1. *Expansion of decycling set for* $m \equiv 1 \pmod 6$, $n \equiv 1 \pmod 3$.

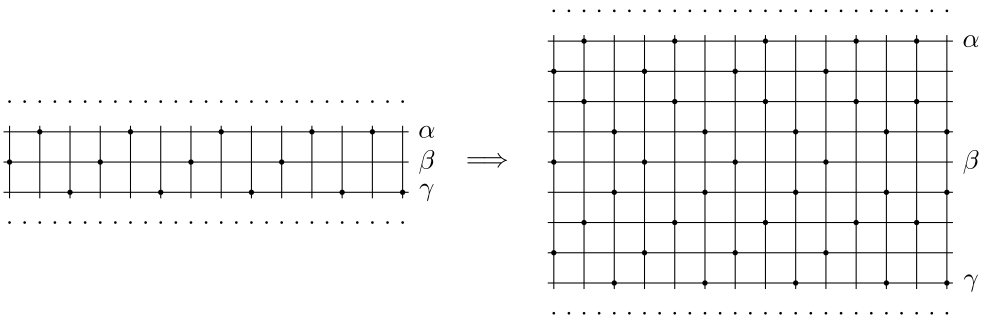


FIG. 3.2. *Expansion of decycling set for* $m \equiv 1 \pmod 6$, $n \equiv 2 \pmod 3$.

row of $C_7\square C_n$ (i.e., row j is type-α if the set of vertices removed from it is $\bigcup_{i=1}^{t}\{v_{j,3i-1}\}\cup\{v_{j,n-1}\}$, where $n = 3t+2$). Similarly, type-β (resp., type-γ) rows are those with a configuration identical to that of the fifth ($\bigcup_{i=1}^{t}\{v_{j,3i-2}\}$) (resp., sixth ($\bigcup_{i=1}^{t}\{v_{j,3i}\}\cup\{v_{j,3t+2}\}$)) row of $C_7\square C_n$.

Focusing on the three consecutive rows that are, in order, of type-α, β, γ in $C_{7+6k}\square C_n$, for some $k \geqslant 0$, we insert six new rows and obtain a minimum decycling set of $C_{7+6(k+1)}\square C_n$. Following the row of type-α in $C_{7+6k}\square C_n$, insert three new rows, being of type-β, type-α, and type-γ in that order, and now following the original type-β row, insert another three new rows, being of type-γ, type-α, and type-β in order (see Figure 3.2 for the expansion pattern where $n = 14$). We now have the decycling set of $C_{7+6(k+1)}\square C_n$.

Note that the new graph, $C_{7+6(k+1)}\square C_n$, contains three consecutive rows that are of type-α, β, γ, so we can iterate the expansion procedure.

In each case we obtain a decycling set S of size $2rn + \lceil\frac{n+2}{3}\rceil$ in $C_m\square C_n$. Moreover, the iteration procedure does not contribute to $\varepsilon(S)$, and hence we conclude that $\omega((C_m\square C_n) - S) = 1$. □

LEMMA 3.2. *If* $m = 6r+5$, *then* $\nabla(C_m\square C_n) = 2rn + \lceil\frac{5n+2}{3}\rceil$.

Proof. Similar to Lemma 3.1, we have the lower bound of the decycling set, $\nabla(C_{6r+5}\square C_n) = 2rn + \lceil\frac{5n+2}{3}\rceil$. We can start from the decycling set for $C_5\square C_n$ described in Lemma 2.9, and insert $6r$ rows of vertices to get the decycling set for $C_{6r+5}\square C_n$. We divide this case into 2 subcases.

(i) $n \equiv 1 \pmod 3$.

Starting from the decycling set of $C_5\square C_n$, a row is type-α if its deleted vertices are in the same columns as those of the third row of $C_5\square C_{3t+1}$ (i.e., row j is type-α if the set of vertices removed from it is $\bigcup_{i=1}^{t+1}\{v_{j,3i-2}\}$), where $n = 3t + 1$. Similarly, type-β (resp., type-γ) rows are those with a configuration identical to that of the fourth $(\bigcup_{i=1}^{t}\{v_{j,3i-1}\} \cup \{v_{j,n}\})$ (resp., fifth $(\bigcup_{i=1}^{t}\{v_{j,3i}\})$) row of $C_5\Box C_{3t+1}$.

Focus on three consecutive rows that are, in order, of type-α, β, γ in $C_{5+6k}\Box C_{3t+1}$ for some $k \geqslant 0$. We now insert three new rows following the type-α row in $C_{5+6k}\Box C_{3t+1}$, the first two being of type-β and type-γ, respectively. For the third of these new rows, add the vertices in columns $3i - 2$ $(i = 1, 2, \dots, t)$ to the decycling set.

Then following the original type-β row, insert another three new rows. For the first of these rows, we select the vertices in columns $3i+1$ $(i = 1, 2, \dots, t-1)$ and $3t$ to add to the decycling set. For the second row, we select the vertices in columns 1 and $3i$ $(i = 1, 2, \dots, t - 1)$. For the third row, we delete vertices so that it is a type-β row (see Figure 3.3 for the expansion pattern where $n = 13$). We now have a minimum decycling set of $C_{5+6(k+1)}\Box C_{3t+1}$.

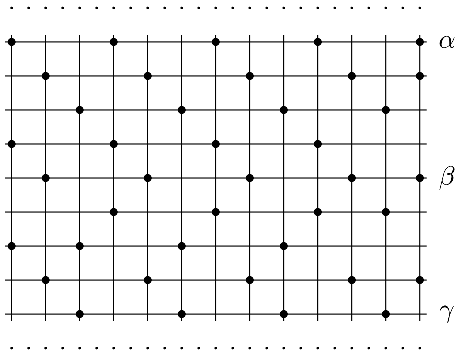


FIG. 3.3. *Expansion result for* $m \equiv 5 \pmod 6$, $n \equiv 1 \pmod 3$.

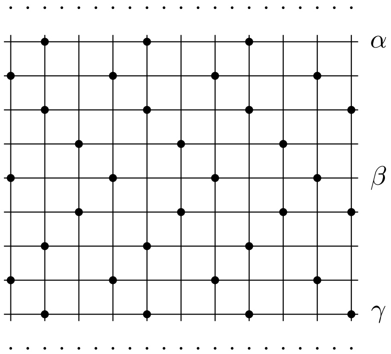


FIG. 3.4. *Expansion result for* $m \equiv 5 \pmod 6$, $n \equiv 2 \pmod 3$.

(ii) $n \equiv 2 \pmod 3$.

Beginning with the decycling set of $C_5\Box C_n$, a row is type-α if its deleted vertices are in the same columns as those of the second row of $C_5\Box C_{3t+2}$ which was described in Lemma 2.9 (i.e., row j is type-α if the set of vertices removed from it is $\bigcup_{i=1}^{t}\{v_{j,3i-1}\}$ where $n = 3t + 2$). Similarly, type-β (resp., type-γ) rows are those with a configuration identical to that of the third $(\bigcup_{i=1}^{t+1}\{v_{j,3i-2}\})$ (resp., fourth $(\bigcup_{i=1}^{t+1}\{v_{j,3i-1}\})$) row of $C_5\Box C_{3t+2}$.

Focus on any three consecutive rows that are, in order, of type-α, β, γ in $C_{5+6k}\Box C_n$, for some $k \geqslant 0$. Following the type-α row, insert three new rows, the first two being type-β and type-γ, respectively. For the third of these new rows, add the vertices in columns $3i$ (where $i = 1, 2, \dots, t$) to the decycling set.

Then, following the original type-β row, insert another three new rows. For the first of these rows, we select the vertices in columns $3i$ (where $i = 1, 2, \dots, t$) and n to add to the decycling set. For the remaining two rows, delete vertices so that they are of type-α and type-β in order (see Figure 3.4 for the expansion pattern where $n = 11$). We now have a minimum decycling set for $C_{5+6(k+1)}\Box C_n$.

Note that in both cases, the new graph, $C_{5+6(k+1)}\Box C_n$, contains three consecutive rows that are of type-α, β, γ, and therefore we can iterate the expansion procedure to obtain a decycling set S of size $2rn+\lceil\frac{5n+2}{3}\rceil$in $C_m\Box C_n$. Moreover $\omega((C_m\Box C_n)-S) =$

1 and so $(C_m \square C_n) - S$ is a maximum induced tree in $C_m \square C_n$. □

For the remaining cases, we use another method to deal with them.

LEMMA 3.3. *If* $m \equiv 2$ *or* 4 (mod 6) *and* $n \equiv 2$ *or* 4 (mod 6)*, then* $\nabla(C_m \square C_n) = \lceil \frac{mn+2}{3} \rceil$.

Proof. For these cases, both m and n are even, so we use the similar method described in Lemma 2.8. Let k be the least nonnegative integer such that at least one of $\frac{m}{2^k}$ or $\frac{n}{2^k}$ is odd or equals 8, and let $m_i = \frac{m}{2^{k-i}}$ and $n_i = \frac{n}{2^{k-i}}$ for each $i = 0, 1, \ldots, k$. Then we can find a minimum decycling set S_0 of cardinality $\lceil \frac{m_0 n_0 + 2}{3} \rceil$ in $C_{m_0} \square C_{n_0}$ such that $\omega\big((C_{m_0} \square C_{n_0}) - S_0\big) = 1$. Now, for each $i = 0, 1, \ldots, k-1$, apply Theorem 2.5 to $C_{m_i} \square C_{n_i}$ to construct a minimum decycling set S_{i+1} of size $\nabla(C_{m_0} \square C_{n_0}) + \frac{mn}{2^{2k}} \sum_{j=0}^{i} 4^j$ in $C_{m_{i+1}} \square C_{n_{i+i}}$ such that $\omega((C_{m_{i+1}} \square C_{n_{i+1}}) - S_{i+1}) = 1$. It follows that $\nabla(C_m \square C_n) = \nabla(C_{m_0} \square C_{n_0}) + \frac{mn}{2^{2k}} \sum_{j=0}^{k-1} 4^j = \lceil \frac{mn+2}{3} \rceil$. □

We have now completely solved not only the problem of finding a minimum decycling set in $C_m \square C_n$, but also the problem of finding a maximum induced tree. We summarize the cardinalities of each set of vertices.

THEOREM 3.4. *Let* $m \geqslant 3$ *and* $n \geqslant 3$ *be integers. Then*

$$\nabla(C_m \square C_n) = \begin{cases} \left\lceil \dfrac{3n}{2} \right\rceil & \textit{if } m = 4, \\ \left\lceil \dfrac{3m}{2} \right\rceil & \textit{if } n = 4, \\ \left\lceil \dfrac{mn+2}{3} \right\rceil & \textit{otherwise} \end{cases}$$

and

$$t(C_m \square C_n) = \begin{cases} 9 & \textit{if } m = n = 4, \\ mn - \nabla(C_m \square C_n) & \textit{otherwise.} \end{cases}$$

4. Remarks.

By removing the row and column which have the maximum number of vertices in the minimum decycling set of $C_{m+1} \square C_{n+1}$, we obtain a (not necessarily minimum) decycling set for $P_m \square P_n$. Thus we have the following corollary.

COROLLARY 4.1. *Let* $m, n > 3$ *be integers. Then*

$$\nabla(P_m \square P_n) \leqslant \begin{cases} \left\lceil \dfrac{(m+1)(n+1)+2}{3} \right\rceil - \dfrac{m+1}{2} - \dfrac{n+1}{2} & \textit{if both } m \textit{ and } n \textit{ are odd,} \\ \left\lceil \dfrac{(m+1)(n+1)+2}{3} \right\rceil - \left\lceil \dfrac{m+1}{3} \right\rceil - \left\lceil \dfrac{n+1}{3} \right\rceil & \textit{otherwise.} \end{cases}$$

Proof. If both m and n are odd, we apply the technique described in Theorem 2.5. Consider a minimum decycling set for $C_{\frac{m+1}{2}} \square C_{\frac{n+1}{2}}$, then use the double construction technique to get a decycling set for $C_{m+1} \square C_{n+1}$ in which there exist a row and a column such that half of the vertices in those row and column are in the decycling set of $C_{m+1} \square C_{n+1}$. Hence $\nabla(P_m \square P_n) \leqslant \nabla(C_{m+1} \square C_{n+1}) - \frac{m+1}{2} - \frac{n+1}{2}$.

Otherwise, in the minimum decycling set for $C_{m+1} \square C_{n+1}$ which was described in sections 2 and 3, we can select a row and a column, one third of the vertices of which are deleted for decycling. □

We observe that the upper bound for $\nabla(P_m \square P_n)$ obtained in Corollary 4.1 is comparable to the upper bound of Beineke and Vandell [5, Theorem 5.4], and when m and n are both odd, it is as good as the upper bound obtained by Caragiannis, Kaklamanis, and Kanellopoulos [7, Theorem 6].

REFERENCES

[1] J. Akiyama and M. Watanabe, *Maximum induced forests of planar graphs*, Graphs Combin., 3 (1987), pp. 201–202.

[2] M. O. Albertson and D. Berman, *A conjecture on planar graphs*, in Graph Theory and Related Topics, J. A. Bondy and U. S. R. Murty, eds., Academic Press, New York, 1979, p. 357.

[3] S. Bau and L. W. Beineke, *The decycling number of graphs*, Australas. J. Combin., 25 (2002), pp. 285–298.

[4] S. Bau, L. W. Beineke, G. Du, Z. Liu, and R. C. Vandell, *Decycling cubes and grids*, Util. Math., 59 (2001), pp. 129–137.

[5] L. W. Beineke and R. C. Vandell, *Decycling graphs*, J. Graph Theory, 25 (1997), pp. 59–77.

[6] J. A. Bondy, G. Hopkins, and W. Staton, *Lower bounds for induced forests in cubic graphs*, Canad. Math. Bull., 30 (1987), pp. 193–199.

[7] I. Caragiannis, C. Kaklamanis, and P. Kanellopoulos, *New bounds on the size of the minimum feedback vertex set in meshes and butterflies*, Inform. Process. Lett., 83 (2002), pp. 275–280.

[8] P. Erdős, M. Saks, and V. T. Sós, *Maximum induced trees in graphs*, J. Combin. Theory Ser. B, 41 (1986), pp. 61–79.

[9] P. Festa, P. M. Pardalos, and M. G. C. Resende, *Feedback set problems*, in Handbook of Combinatorial Optimization, Supplement Vol. A, Kluwer, Dordrecht, 1999, pp. 209–258.

[10] E. Speckenmeyer, *Bounds on feedback vertex sets of undirected cubic graphs*, in Algebra, Combinatorics and Logic in Computer Science, Vol. I,II (Győr, 1983), Colloq. Math. Soc. János Bolyai 42, North-Holland, Amsterdam, 1986, pp. 719–729.

[11] R. J. Wilson, *Introduction to Graph Theory*, 4th ed., Addison-Wesley/Longman, Reading, MA, 1996.

SIAM J. DISCRETE MATH.
Vol. 19, No. 3, pp. 664–690

ISOMORPH-FREE EXHAUSTIVE GENERATION OF DESIGNS WITH PRESCRIBED GROUPS OF AUTOMORPHISMS*

PETTERI KASKI†

Abstract. We develop an algorithm framework for isomorph-free exhaustive generation of designs admitting a group of automorphisms from a prescribed collection of pairwise nonconjugate groups, where each prescribed group has a large index relative to its normalizer in the isomorphism-inducing group. We demonstrate the practicality of the framework by producing a complete classification of the Steiner triple systems of order 21 admitting a nontrivial automorphism group. The number of such pairwise nonisomorphic designs is 62336617, where 958 of the designs are anti-Pasch. We also develop consistency checking methodology for gaining confidence in the correct operation of the algorithm implementation.

Key words. classification algorithm, combinatorial search, consistency checking, isomorph rejection, isomorph-free generation, Kramer–Mesner method, Steiner triple system, symmetry reduction

AMS subject classifications. 05-04, 68R05, 05B07, 51E10

DOI. 10.1137/S0895480104444788

1. Introduction.

Among the basic problems in combinatorics is the classification of various types of combinatorial designs [2, 12] that admit a prescribed group of automorphisms. For small to intermediate parameter values, such classification problems can be studied using computer search in the following framework.

Let Π be a finite set of *points* and let G be a finite group that acts on Π. Let Ω be a set of subsets of Π that is closed under the induced action of G; the subsets in Ω are called *blocks*. A *set system* $\mathcal{X}$ is a multiset of blocks.

Two set systems $\mathcal{X}, \mathcal{Y}$ are *isomorphic* if they are on the same orbit under the induced action of G. A group element $g \in G$ that satisfies $g\mathcal{X} = \mathcal{Y}$ is an *isomorphism* of $\mathcal{X}$ onto $\mathcal{Y}$. The group $\mathrm{Aut}(\mathcal{X}) = \{g \in G : g\mathcal{X} = \mathcal{X}\}$ is the *automorphism group* of $\mathcal{X}$. A subgroup of $\mathrm{Aut}(\mathcal{X})$ is a *group of automorphisms*.

Let P be an isomorphism invariant structural property for set systems (such as the property of being a design), and let H be a subgroup of G. The associated *classification problem* is to generate exactly one set system $\mathcal{X}$ from every isomorphism class that satisfies property P and admits (a group conjugate to) H as a group of automorphisms. More generally, if $\mathcal{H}$ is a set of pairwise nonconjugate subgroups of G, then the task is to generate up to isomorphism all set systems $\mathcal{X}$ that satisfy P and admit (a group conjugate to) at least one of the groups in $\mathcal{H}$ as a group of automorphisms.

The celebrated Kramer–Mesner method [39] and methods based on tactical decompositions [18, 19] are prominent examples of computational methods for attacking classification problems under prescribed groups of automorphisms; see, for example, [3, 4, 5, 29, 40, 41, 42, 45, 46, 64, 73] and [13, 17, 30, 38, 50, 59, 60, 67, 71, 72], respectively, and the survey articles [28, 51]. Other recent classification techniques include [25, 35, 49].

*Received by the editors July 1, 2004; accepted for publication (in revised form) February 28, 2005; published electronically November 23, 2005.
http://www.siam.org/journals/sidma/19-3/44478.html

†Laboratory for Theoretical Computer Science, Helsinki University of Technology, P.O. Box 5400, FI-02015 TKK, Finland (petteri.kaski@hut.fi).

In this paper we focus on the classification of set systems with a prescribed group of automorphisms H such that H has a large index relative to the normalizer $N_G(H) = \{g \in G : gHg^{-1} = H\}$, and the kernel of the induced action of $N_G(H)$ on the orbits $H\backslash\Omega$ is either H or a small overgroup of H. This situation occurs recurrently when we are interested in classifying up to isomorphism all systems with a nontrivial automorphism group; see, for example, [13, 35, 49, 56, 65]. In such applications, the set $\mathcal{H}$ contains—up to conjugacy in G—all eligible prime-order subgroups of G, some of which can have very large normalizers.

To obtain a practical isomorph-free exhaustive generation algorithm in the large-index normalizer case, we must address two primary difficulties. First, we must eliminate the symmetry induced by $N_G(H)$ from the search space. (If no symmetry reduction is performed, we potentially end up generating a very large number of isomorphic systems, which results in an impractical algorithm. To give an intuitive justification to this claim, observe that if $\mathcal{X}$ is a set system with $H \leq \mathrm{Aut}(\mathcal{X})$, then $g\mathcal{X}$ is an isomorphic set system with $H \leq \mathrm{Aut}(g\mathcal{X})$ for all $g \in N_G(H)$.) Second, we must reject G-isomorphs among the generated set systems to obtain isomorph-free generation. An extensive discussion of isomorph rejection in classification algorithms occurs in [6, 26, 45, 46, 55, 61].

The main contribution of this paper is a classification framework that extends the "generation via seeds" approach employed in [31, 32, 33] to the setting of prescribed automorphism groups. Alternatively, the present contribution can be seen as extending the traditional Kramer–Mesner method [39] by a symmetry reduction front-end and an isomorph rejection back-end.

First, we develop a general technique for performing symmetry reduction in the prescribed automorphism group setting. The technique is based on classifying up to $N_G(H)$-isomorphism a collection of subsystems—called seeds—such that every set system $\mathcal{X}$ satisfying P and $H \leq \mathrm{Aut}(\mathcal{X})$ must contain at least one seed as a subsystem. We then achieve exhaustive generation by computing all possible extensions of every seed to a complete system. Here a number of algorithms can be employed depending on the type of design being classified.

Second, to achieve isomorph-free generation, we develop an isomorph rejection technique compatible with the "generation via seeds" approach. The technique is based on the canonical construction path method [55]. Neither isomorphism testing between generated objects nor keeping a record of the generated objects is required, which enables parallelization and classification of families with millions or even billions of nonisomorphic designs. The applicability of the technique is, however, limited by the need to perform isomorphism computations (such as computing canonical labeling and generators for the automorphism group), which can become prohibitively expensive as the size of the generated systems increases. In this connection we wish to mention the recent group-theoretic isomorph rejection technique developed in [46], which altogether avoids isomorphism computations; however, this technique appears not to be applicable in the large-index normalizer setting, because knowledge of all the subgroups in the interval $[H, N_G(H)]$ in the subgroup lattice of G is required.

As a secondary contribution, to illustrate the applicability of the framework, we produce a complete classification for Steiner triple systems of order 21 with a nontrivial automorphism group. For order 21, a complete classification was previously known only for certain prime-order automorphisms [15, 52, 70] and Kirkman systems with a nontrivial automorphism group [11].

This paper is organized as follows. Section 2 begins with some definitions and notation. The classification framework is described in three parts. Section 3 gives a top-level description of the framework and proves its correctness under certain assumptions on the seeds. Section 4 describes the construction procedure for the seeds and proves its correctness. Section 5 discusses the implementation of nontrivial subroutines—such as canonical labeling algorithms—required by the framework.

Section 6 contains the results obtained for Steiner triple systems of order 21 admitting a nontrivial automorphism group. Section 7 implements consistency checking techniques for guarding against errors in algorithm implementation. The paper is concluded in section 8 with a discussion of further applications and limitations of the framework.

2. Definitions and notation.

2.1. Groups and group actions. For the background in groups and group actions, see [23, 34, 63]. In this paper groups act from the left ("$g\mathcal{X}$") and permutations compose from right to left ("$(1\ 2)(2\ 3) = (1\ 2\ 3)$"). We write S_n for the symmetric group of degree n and $\mathrm{Sym}(\Delta)$ for the symmetric group on a finite set Δ.

Let G, Π, Ω be as defined in the introduction. We write $H \leq G$ to indicate that H is a subgroup of G. The action of G on Π and Ω induces by restriction an action of $H \leq G$. We write $\mathrm{fix}(H)$ for the set of points fixed by the action of H on Π, and $H\backslash\Omega$ for the set of all H-orbits on Ω. We write $\mathcal{X} \subseteq \Omega$ to indicate that $\mathcal{X}$ is a set system, and $\mathcal{X} \subseteq_H \Omega$ to indicate that $\mathcal{X}$ is a set system with $H \leq \mathrm{Aut}(\mathcal{X})$.

An element $g \in G$ acts on a set system $\mathcal{X} \subseteq \Omega$ by acting on the elements occurring in the blocks $B \in \mathcal{X}$. More precisely, the set system $g\mathcal{X}$ is defined by the following rule: a block B occurs in $\mathcal{X}$ with multiplicity m if and only if the block $gB = \{gx : x \in B\}$ occurs in $g\mathcal{X}$ with multiplicity m.

An element $g \in G$ acts on $H \leq G$ by conjugation, that is, the *g-conjugate* of H is the subgroup $gHg^{-1} = \{ghg^{-1} : h \in H\}$. Two subgroups $H_1, H_2 \leq G$ are *conjugate* if they are on the same orbit under the conjugation action on the set of all subgroups of G.

For a subset $S \subseteq G$, we write $\langle S \rangle$ for the subgroup of G generated by S.

The following notation will be convenient in discussing the seeds associated with the classification algorithm. Let $H \leq G$, $T \subseteq \Pi$, and $\mathcal{X} \subseteq_H \Omega$. We write $(H, T)\!\downarrow\!\mathcal{X}$ for the union of those H-orbits in $\mathcal{X}$ that contain at least one block that has nonempty intersection with T.

2.2. Isomorphism and canonical labeling. An extensive discussion of isomorphism and symmetry appears in [1]. Let $K \leq G$. Two set systems $\mathcal{X}_1, \mathcal{X}_2 \subseteq \Omega$ are *K-isomorphic* if there exists a $g \in K$ such that $\mathcal{X}_2 = g\mathcal{X}_1$. Such an element g is a *K-isomorphism* from $\mathcal{X}_1$ to $\mathcal{X}_2$. We write $\mathcal{X}_1 \cong_K \mathcal{X}_2$ to indicate that $\mathcal{X}_1$ and $\mathcal{X}_2$ are K-isomorphic. If $G = K$, then we omit the group symbol K from the notation. The *K-automorphism group* of $\mathcal{X}$ is

$$\mathrm{Aut}_K(\mathcal{X}) = \{g \in K : g\mathcal{X} = \mathcal{X}\}.$$

We use similar notation for indicating isomorphism of tuples of objects under the induced action of G. For example, if $\mathcal{X}_1, \mathcal{X}_2 \subseteq \Omega$ and $T_1, T_2 \subseteq \Pi$, then we write $(\mathcal{X}_1, T_1) \cong_K (\mathcal{X}_2, T_2)$ to indicate that there exists a $g \in K$ such that $g\mathcal{X}_1 = \mathcal{X}_2$ and $gT_1 = T_2$. Similarly, we denote by $\mathrm{Aut}_K(\mathcal{X}_1, T_1)$ the automorphism group $\{g \in K : g\mathcal{X}_1 = \mathcal{X}_1 \text{ and } gT_1 = T_1\}$.

A *K-canonical labeling map* κ associates to each $\mathcal{X} \subseteq \Omega$ a group element $\kappa(\mathcal{X}) \in K$ such that $\kappa(g\mathcal{X})g\mathcal{X} = \kappa(\mathcal{X})\mathcal{X}$ holds for all $g \in K$. The set system $\kappa(\mathcal{X})\mathcal{X}$ is the *K-canonical representative* of the K-orbit of $\mathcal{X}$.

2.3. Designs. A *Steiner triple system* of order v—briefly, an STS(v)—is a set system consisting of v points and $v(v-1)/6$ blocks of size three such that every pair of distinct points occurs in a unique block.

An STS(v) exists if and only if either $v \equiv 1 \pmod 6$ or $v \equiv 3 \pmod 6$. An excellent reference to Steiner triple systems is [14].

3. The classification framework.

In this section we give a top-level description of the classification framework, which we then proceed to refine in subsequent sections. Examples illustrate the implementation of the algorithm for our subsequent application of classifying the Steiner triple systems of order 21 with a nontrivial automorphism group.

Suppose G, Π, Ω, and P have been fixed, and let $\mathcal{H}$ be a nonempty set of pairwise nonconjugate subgroups of G.

The goal of a classification algorithm based on the framework is to generate up to isomorphism all set systems $\mathcal{X} \subseteq \Omega$ that satisfy the structural property P together with the property that $\mathrm{Aut}(\mathcal{X})$ has at least one subgroup that is conjugate to one of the groups in $\mathcal{H}$. For brevity, we call such set systems *target systems*.

Example 1. To generate Steiner triple systems of order v, let $G = S_v$ act on $\Pi = \{1, 2, \ldots, v\}$, and let Ω consist of all the 3-subsets of Π.

To generate all systems admitting a nontrivial automorphism group, we let $\mathcal{H}$ consist of all eligible pairwise nonconjugate prime-order subgroups of S_v. (Clearly, any nontrivial automorphism group must have at least one prime-order subgroup.) The prime-order subgroups of S_{21} that can occur as a group of automorphisms of an STS(21) can be determined by combinatorial arguments; see section 6. A more detailed treatment appears in [14, Ch. 7].

As outlined in the introduction, the framework achieves exhaustive generation by first determining a set of seeds and then extending the seeds to target systems in all possible ways. Obviously the applicability of the framework depends heavily on whether we can obtain a “suitable” collection of seeds for a given choice of $G, \Pi, \Omega, P, \mathcal{H}$.

Ideally, we would like the seeds associated with an $H \in \mathcal{H}$ to satisfy the following properties:

(a) the seeds should be fast to classify up to $N_G(H)$-isomorphism;

(b) the seeds should have a small $N_G(H)$-automorphism group;

(c) every target system should contain as few seeds as possible; and

(d) given a target system $\mathcal{X}$ and $H_0 \leq \mathrm{Aut}(\mathcal{X})$ conjugate to H, it should be possible to identify the seeds associated with H_0 occurring in $\mathcal{X}$.

Property (a) is obviously required. Property (b) is related to symmetry reduction. To illustrate the role of the automorphism group, consider the following situation. Suppose $\mathcal{S} \subseteq_H \Omega$, and let $\mathcal{X}$ be a target system that extends $\mathcal{S}$; that is, $H \leq \mathrm{Aut}(\mathcal{X})$ and $\mathcal{S} \subseteq \mathcal{X}$. The extension phase generates all such extensions of $\mathcal{S}$. In particular, if $\mathcal{X}$ extends $\mathcal{S}$, then so does $g\mathcal{X}$ for all $g \in \mathrm{Aut}_{N_G(H)}(\mathcal{S})$. Thus, the smaller the automorphism group $\mathrm{Aut}_{N_G(H)}(\mathcal{S})$, the fewer isomorphic extensions of $\mathcal{S}$ must be generated. Property (c) is also related to reducing the number of occurrences of isomorphic target systems. Namely, by the structure of the framework, an isomorphism class of target systems is generated at least once as an extension of every isomorphism class of seeds occurring in it. Property (d) is related to isomorph rejection on target systems. In particular, we must be able to identify from every target system a “parent seed” from which the target system must originate. Furthermore, for purposes of consistency checking it is useful to be able to identify every seed occurring in a target system.

In many cases we can obtain a good collection of seeds by selecting a set of points $T \subseteq \Pi$ and considering a subsystem "induced" by the pair H, T in a target system $\mathcal{X}$ with $H \leq \mathrm{Aut}(\mathcal{X})$. For example, we can consider as a seed the H-orbits in $\mathcal{X}$ containing a block that intersects T. With an appropriate choice of T, this often results in a good collection of seeds for designs with pairwise balance constraints. For t-designs with $t > 2$, we can consider a derived design induced by T, $|T| < t$; cf. [35]. In what follows we use the former type of seed.

3.1. The parent seeds. We begin by describing the procedure that we use to distinguish the parent seed from a target system. This procedure consists of a sequence of distinguishing operations.

Given a target system $\mathcal{X}$, we first distinguish a subgroup $H_0 \leq \mathrm{Aut}(\mathcal{X})$ that is conjugate to one of the groups in $\mathcal{H}$. This distinguishing operation is canonical in the following sense:

(3.1) If H_0 and H_0' are the subgroups distinguished from $\mathcal{X}$ and $\mathcal{X}'$, respectively, and $\mathcal{X} \cong \mathcal{X}'$, then $(\mathcal{X}, H_0) \cong (\mathcal{X}', H_0')$.

Here and in what follows G acts on its subgroups by conjugation.

Example 2. In our applications, the distinguishing operation is implemented as follows. Let κ be a G-canonical labeling map. (Implementation of the canonical labeling maps is discussed in section 5.) Order the subgroups of G arbitrarily (for example, order the elements in G and use lexicographic order for subsets of G to obtain an ordering for the subgroups). Given a target system $\mathcal{X}$, put

$$\begin{aligned} &\hat{\mathcal{X}} \leftarrow \kappa(\mathcal{X})\mathcal{X}, \\ &\hat{H}_0 \leftarrow \min\{\hat{H} \leq \mathrm{Aut}(\hat{\mathcal{X}}) : \hat{H} \text{ is conjugate to a group in } \mathcal{H}\}, \\ &H_0 \leftarrow \kappa(\mathcal{X})^{-1}\hat{H}_0\kappa(\mathcal{X}). \end{aligned} \tag{3.2}$$

In our applications, the groups in $\mathcal{H}$ are prime-order cyclic groups, so to evaluate (3.2) relative to the lexicographic order for subgroups, it suffices to find—relative to some ordering of the elements in G—the minimum nonidentity element $\hat{h}_0$ in $\mathrm{Aut}(\hat{\mathcal{X}})$ that belongs to a conjugacy class that intersects a group in $\mathcal{H}$. We can then put $\hat{H}_0 \leftarrow \langle\{\hat{h}_0\}\rangle$. Because $\mathrm{Aut}(\hat{\mathcal{X}})$ in most cases has a relatively small order, this implementation suffices for our purposes.

Then, based on $\mathcal{X}$ and H_0, we distinguish a nonempty set $T_0 \subseteq \Pi$ of points. Also this distinguishing operation is canonical in the following sense:

(3.3) If T_0 and T_0' are distinguished from $(\mathcal{X}, H_0)$ and $(\mathcal{X}', H_0')$, respectively, and $(\mathcal{X}, H_0) \cong (\mathcal{X}', H_0')$, then $(\mathcal{X}, H_0, T_0) \cong (\mathcal{X}', H_0', T_0')$.

The precise structure of the distinguishing operation for T_0 depends on the selected transversal invariants—which will be discussed in section 4.2. Here are two examples that illustrate the possibilities.

Example 3. Let $f \leq k$ be nonnegative integers. Order the subsets of Π arbitrarily. Assuming that $\hat{H}_0$ has been computed as in (3.2), put

$$\begin{aligned} &\hat{T}_0 \leftarrow \min\{\hat{T} \subseteq \Pi : |\hat{T}| = k \text{ and } |\hat{T} \cap \mathrm{fix}(\hat{H}_0)| = f\}, \\ &T_0 \leftarrow \kappa(\mathcal{X})^{-1}\hat{T}_0. \end{aligned} \tag{3.4}$$

Here we must be careful that $f \leq |\mathrm{fix}(\hat{H}_0)|$ and $k - f \leq |\Pi - \mathrm{fix}(\hat{H}_0)|$.

Example 4. To achieve properties (a)–(d) discussed in the beginning of this section, it is useful to further constrain the possible sets $\hat{T}$ in (3.4). For example, we can require that

(3.5) $\hat{T}$ must occur as a (subset of a) block in $\hat{\mathcal{X}}$, and/or

(3.6) no two points in $\hat{T}$ occur in the same $\hat{H}_0$-orbit.

Again we must be careful that a set $\hat{T}_0$ satisfying the additional requirements actually exists in every target system $\hat{\mathcal{X}}$ with $\hat{H}_0 \leq \mathrm{Aut}(\hat{\mathcal{X}})$.

Finally, based on $\mathcal{X}$, H_0, and T_0, we let $\mathcal{S}_0$ be the union of those H_0-orbits on $\mathcal{X}$ that contain at least one block that has nonempty intersection with T_0. In other words, we put $\mathcal{S}_0 \leftarrow (H_0, T_0) \downarrow \mathcal{X}$. The three-tuple $(H_0, T_0, \mathcal{S}_0)$ is the *parent seed* associated with $\mathcal{X}$.

From (3.1), (3.3), and the definition of "$\downarrow$" it follows that

(3.7) if $(H_0, T_0, \mathcal{S}_0)$ and $(H_0', T_0', \mathcal{S}_0')$ are the parent seeds associated with $\mathcal{X}$ and $\mathcal{X}'$, respectively, and $\mathcal{X} \cong \mathcal{X}'$, then $(\mathcal{X}, H_0, T_0, \mathcal{S}_0) \cong (\mathcal{X}', H_0', T_0', \mathcal{S}_0')$.

3.2. The top-level algorithm. The top-level classification algorithm is executed for each group $H \in \mathcal{H}$ in turn. The execution of the algorithm is divided into three stages.

3.2.1. Seed generation. The first stage in the algorithm is a backtrack search that takes as input the group $H \in \mathcal{H}$, and produces a set of pairwise nonisomorphic three-tuples of the form $(H, T, \mathcal{S})$, where $T \subseteq \Pi$ and $\mathcal{S} \subseteq_H \Omega$. Each such three-tuple output by the algorithm is called an H-*seed* (or simply a *seed* if the group H is clear from the context). The set of H-seeds has the following property:

(3.8) For every target system $\mathcal{X}$ and the associated parent seed $(H_0, T_0, \mathcal{S}_0)$, either $(H_0, T_0, \mathcal{S}_0)$ is isomorphic to a unique H-seed $(H, T, \mathcal{S})$ or H_0 is not conjugate to H.

The seed generation algorithm will be described in more detail in section 4.

3.2.2. Seed extension. The second stage takes as input a seed $(H, T, \mathcal{S})$ and constructs all target systems $\mathcal{X}$ such that $H \leq \mathrm{Aut}(\mathcal{X})$ and $(H, T) \downarrow \mathcal{X} = \mathcal{S}$; such target systems are said to *extend* $(H, T, \mathcal{S})$. Depending on the property P, a number of algorithms can be employed in the extension stage. An approach that is often practical is to transform the extension problem into a well-known combinatorial problem, and then apply algorithms developed for this problem.

Example 5. The problem of extending a seed $(H, T, \mathcal{S})$ into an STS(v) is equivalent to an instance of the exact cover problem, in which the task is to find all possible ways to cover the H-orbits of 2-subsets of Π not already covered by $\mathcal{S}$ using H-orbits of 3-subsets. A state-of-the-art algorithm for the exact cover problem appears in [37].

Example 6. *Other standard problems that are often applicable include clique problems* [58] *and the problem of determining all solutions to a Diophantine linear system of equations with upper and lower bounds on the variables* [73].

3.2.3. Isomorph rejection. The third stage of the algorithm takes as input a target system $\mathcal{X}$ that extends the seed $(H, T, \mathcal{S})$ and performs two isomorph rejection tests. If $\mathcal{X}$ passes both tests, then it is output as the unique representative of its isomorphism class.

The first test accepts $\mathcal{X}$ if and only if the parent seed $(H_0, T_0, \mathcal{S}_0)$ associated with $\mathcal{X}$ is $\mathrm{Aut}(\mathcal{X})$-isomorphic to $(H, T, \mathcal{S})$. The second test accepts $\mathcal{X}$ if and only if $\mathcal{X}$ is the minimum object from its $\mathrm{Aut}(H, T, \mathcal{S})$-orbit (for example, relative to lexicographic order). Alternatively—in the case $\mathrm{Aut}(H, T, \mathcal{S})$ is too large for a minimality test to be practical—the second test can be implemented using a hash table that contains the canonical representatives for target systems $\mathcal{X}$ encountered earlier as extensions of $(H, T, \mathcal{S})$. The alternative test accepts $\mathcal{X}$ if and only if the canonical representative of $\mathcal{X}$ does not occur in the hash table. (The drawback of this alternative test is that we have to store the canonical representatives encountered, which can be very memory-intensive despite the need to store only the extensions of a single seed at a time.)

Example 7. From $(H, T) \downarrow \mathcal{X} = \mathcal{S}$ and $(H_0, T_0) \downarrow \mathcal{X} = \mathcal{S}_0$ it follows that we have $(H, T, \mathcal{S}) \cong_{\mathrm{Aut}(\mathcal{X})} (H_0, T_0, \mathcal{S}_0)$ if and only if $(H, T) \cong_{\mathrm{Aut}(\mathcal{X})} (H_0, T_0)$; thus, it suffices to check only the latter isomorphism relation in performing the first isomorph rejection test. Because $\mathrm{Aut}(\mathcal{X})$ has a small prime order for the vast majority of the target systems in our applications, we can use exhaustive search on $\mathrm{Aut}(\mathcal{X})$ to decide whether $(H, T) \cong_{\mathrm{Aut}(\mathcal{X})} (H_0, T_0)$.

Example 8. In our applications, most seeds $(H, T, \mathcal{S})$ have a small-order automorphism group. Thus, the minimality of $\mathcal{X}$ in its $\mathrm{Aut}(H, T, \mathcal{S})$-orbit can be tested by exhaustive search on $\mathrm{Aut}(H, T, \mathcal{S})$.

This completes the description of the top-level classification algorithm.

3.3. Correctness. We prove correctness of the top-level algorithm under the assumption that the seed generation algorithm is correct; that is, (3.8) holds for all $H \in \mathcal{H}$.

THEOREM 3.1. *Let $\mathcal{X}$ and $\mathcal{X}'$ be target systems that extend the seeds $(H, T, \mathcal{S})$ and $(H', T', \mathcal{S}')$, respectively. Furthermore, let $(H_0, T_0, \mathcal{S}_0)$ and $(H_0', T_0', \mathcal{S}_0')$ be the associated parent seeds. If $\mathcal{X} \cong \mathcal{X}'$, $(H, T, \mathcal{S}) \cong_{\mathrm{Aut}(\mathcal{X})} (H_0, T_0, \mathcal{S}_0)$, and $(H', T', \mathcal{S}') \cong_{\mathrm{Aut}(\mathcal{X}')} (H_0', T_0', \mathcal{S}_0')$, then $(\mathcal{X}, H, T, \mathcal{S}) \cong (\mathcal{X}', H', T', \mathcal{S}')$.*

Proof. From $\mathcal{X} \cong \mathcal{X}'$ and (3.7) we obtain $(\mathcal{X}, H_0, T_0, \mathcal{S}_0) \cong (\mathcal{X}', H_0', T_0', \mathcal{S}_0')$. Thus, it follows from $(H, T, \mathcal{S}) \cong_{\mathrm{Aut}(\mathcal{X})} (H_0, T_0, \mathcal{S}_0)$ and $(H', T', \mathcal{S}') \cong_{\mathrm{Aut}(\mathcal{X}')} (H_0', T_0', \mathcal{S}_0')$ that $(\mathcal{X}, H, T, \mathcal{S}) \cong (\mathcal{X}', H', T', \mathcal{S}')$. □

Theorem 3.1 implies that isomorphic $\mathcal{X}, \mathcal{X}'$ that both pass the first isomorph rejection test must be extensions of isomorphic seeds: $(H, T, \mathcal{S}) \cong (H', T', \mathcal{S}')$. Because seeds output by the seed generation algorithm are pairwise nonisomorphic, we must have $(H, T, \mathcal{S}) = (H', T', \mathcal{S}')$. Consequently, the conclusion of Theorem 3.1 can be written in the stronger form $\mathcal{X} \cong_{\mathrm{Aut}(H,T,\mathcal{S})} \mathcal{X}'$. By the structure of the second isomorph rejection test, this implies that at most one system $\mathcal{X}$ from every isomorphism class of target systems is output.

THEOREM 3.2. *For every target system $\mathcal{X}'$, there exists a target system $\mathcal{X}$ and a seed $(H, T, \mathcal{S})$ such that $\mathcal{X} \cong \mathcal{X}'$, $\mathcal{X}$ extends $(H, T, \mathcal{S})$, and $(H, T, \mathcal{S}) \cong_{\mathrm{Aut}(\mathcal{X})} (H_0, T_0, \mathcal{S}_0)$, where $(H_0, T_0, \mathcal{S}_0)$ is the parent seed associated with $\mathcal{X}$.*

Proof. Let $(H_0', T_0', \mathcal{S}_0')$ be the parent seed associated with $\mathcal{X}'$. By (3.8), there exists a seed $(H, T, \mathcal{S})$ that satisfies $(H, T, \mathcal{S}) \cong (H_0', T_0', \mathcal{S}_0')$. Select any $g' \in G$ that takes $(H_0', T_0', \mathcal{S}_0')$ to $(H, T, \mathcal{S})$. Put $\mathcal{X} = g'\mathcal{X}'$. Because $\mathcal{S}_0' \subseteq \mathcal{X}'$ and $\mathcal{X}' \subseteq_{H_0'} \Omega$, we have $\mathcal{S} \subseteq \mathcal{X}$ and $\mathcal{X} \subseteq_H \Omega$. Thus, $\mathcal{X}$ extends $(H, T, \mathcal{S})$. Now $\mathcal{X} \cong \mathcal{X}'$ implies $(\mathcal{X}, H_0, T_0, \mathcal{S}_0) \cong (\mathcal{X}', H_0', T_0', \mathcal{S}_0')$ by (3.7). Select any $g \in G$ that takes $(\mathcal{X}', H_0', T_0', \mathcal{S}_0')$ to $(\mathcal{X}, H_0, T_0, \mathcal{S}_0)$. Then it follows that $g'g^{-1} \in \mathrm{Aut}(\mathcal{X})$ takes $(H_0, T_0, \mathcal{S}_0)$ to $(H, T, \mathcal{S})$. □

Because g' can be any group element taking $(H'_0, T'_0, \mathcal{S}'_0)$ to $(H, T, \mathcal{S})$ in the proof of Theorem 3.2, it follows that all $\mathcal{X}_1 \subseteq \Omega$ in the $\mathrm{Aut}(H, T, \mathcal{S})$-orbit of $\mathcal{X}$ are constructed from $(H, T, \mathcal{S})$ and satisfy $(H, T, \mathcal{S}) \cong_{\mathrm{Aut}(\mathcal{X}_1)} (H_0, T_0, \mathcal{S}_0)$. Consequently, the isomorph rejection stage outputs exactly one system from every isomorphism class of target systems.

4. The seed generation algorithm. Given a group $H \in \mathcal{H}$ as input, the seed generation algorithm produces a set of H-seeds such that every parent seed $(H_0, T_0, \mathcal{S}_0)$ is either isomorphic to exactly one H-seed or H_0 is not conjugate to H.

The seed generation algorithm uses backtrack search to construct the seeds one H-orbit at a time. A (partial) seed is a three-tuple $(H, T, \mathcal{S})$, where $T \subseteq \Pi$ is the set of points completed so far, and $\mathcal{S} \subseteq_H \Omega$ contains the H-orbits in the seed.

We structure the search so that one point at a time is completed. In other words, we keep track of the current point $p \in \Pi$ being completed and add only H-orbits of blocks incident with p, until the point p is complete. Intuitively, we construct a parent seed $(H, T_0, \mathcal{S}_0)$ via a sequence of partial seeds $(H, T, \mathcal{S})$, where $T \subseteq T_0$, $(H, T) \downarrow \mathcal{S}_0 = \mathcal{S}$, and the set T is enlarged one point at a time until $T = T_0$.

To implement this structure, we use two alternating layers of backtrack search: the transversal layer (which selects the next point to be completed and performs isomorph rejection on partial seeds with ℓ completed points) and the point completion layer (which completes a distinguished point p by adding orbits to the partial seed; isomorph rejection is carried out after every added orbit). Isomorph rejection is based on the canonical construction path method [55].

Often the structural property P can be exploited to further constrain the search. Let $P^{\downarrow}$ be an isomorphism invariant property for three-tuples $(H_0, T_0, \mathcal{S}_0)$, where $H_0 \leq G$, $T_0 \subseteq \Pi$, and $\mathcal{S} \subseteq_{H_0} \Omega$, such that every three-tuple $(H_0, T_0, \mathcal{S}_0)$ originating from some target system $\mathcal{X}$ via "$\downarrow$" has property $P^{\downarrow}$. More precisely, a three-tuple *originates* from $\mathcal{X}$ via "$\downarrow$" if $H_0 \leq \mathrm{Aut}(\mathcal{X})$ and $\mathcal{S}_0 = (H_0, T_0) \downarrow \mathcal{X}$.

Example 9. To generate seeds for Steiner triple systems of order v, we choose $P^{\downarrow}$ as follows. A three-tuple $(H, T, \mathcal{S})$ has property $P^{\downarrow}$ if and only if every point $p \in T$ occurs in exactly $r = (v-1)/2$ blocks in $\mathcal{S}$ and every pair of distinct points occurs in at most one block in $\mathcal{S}$.

We require that $P^{\downarrow}$ is hereditary in the following sense:

(4.1) If $(H, T, \mathcal{S})$ has property $P^{\downarrow}$, then $(H, T_1, (H, T_1) \downarrow \mathcal{S})$ has property $P^{\downarrow}$ for all $T_1 \subseteq T$.

Before we describe the algorithm in detail, we state some conventions that apply throughout this section.

Because the input group $H \in \mathcal{H}$ is fixed, we can simplify notation and treat a partial seed as a pair $(T, \mathcal{S})$ and omit the fixed group H. Similarly, we omit the group H and write $T \downarrow \mathcal{S}$ instead of $(H, T) \downarrow \mathcal{S}$.

We perform all isomorphism computations in this section relative to the normalizer $N_G(H)$ unless we explicitly indicate otherwise. Accordingly, we assume that κ is an arbitrary $N_G(H)$-canonical labeling map for set systems.

To simplify the description of the algorithms, we assume that the seed generation algorithm immediately disregards all partial seeds $(T, \mathcal{S})$ that "obviously" cannot be extended to a seed that satisfies $P^{\downarrow}$.

Example 10. To generate seeds for Steiner triple systems of order v, we immediately disregard a partial seed $(T, \mathcal{S})$ if any pair of distinct points occurs in more than one block in $\mathcal{S}$; cf. Example 9.

4.1. The point completion layer. The point completion layer takes as input a three-tuple $(T_0, \mathcal{S}_0, p_0)$, where

$$T_0 \subseteq \Pi, \quad \mathcal{S}_0 \subseteq_H \Omega, \quad T_0 \downarrow \mathcal{S}_0 = \mathcal{S}_0, \quad \text{and} \quad p_0 \in \Pi - T_0.$$

Here $(T_0, \mathcal{S}_0)$ is the current partial seed, and p_0 is the distinguished point to be augmented by adding H-orbits of blocks.

The point completion layer algorithm constructs up to $N_G(H)$-isomorphism all partial seeds that satisfy $P^{\downarrow}$ and can be obtained from $(T_0, \mathcal{S}_0)$ by adding orbits incident with the point p_0.

More precisely, a three-tuple $(T_1, \mathcal{S}_1, p_1)$ is an *extension* of $(T_0, \mathcal{S}_0, p_0)$ if $(T_1, T_1 \downarrow \mathcal{S}_1, p_1) \cong_{N_G(H)} (T_0, \mathcal{S}_0, p_0)$ and $\{p_1\} \downarrow (\mathcal{S}_1 - (T_1 \downarrow \mathcal{S}_1)) = \mathcal{S}_1 - (T_1 \downarrow \mathcal{S}_1)$. The H-orbits in $\mathcal{S}_1 - (T_1 \downarrow \mathcal{S}_1)$ are called *extending orbits*. The *level* of an extension is the number of extending orbits in it. An extension is *complete* if $(T_1 \cup \{p_1\}, \mathcal{S}_1)$ has property $P^{\downarrow}$. Note that an extension can be complete even if it contains no extending orbits.

4.1.1. Algorithm. A partial solution in the point completion layer backtrack search is an extension $(T_1, \mathcal{S}_1, p_1)$ of $(T_0, \mathcal{S}_0, p_0)$. Initially, the algorithm is invoked with $(T_0, \mathcal{S}_0, p_0)$.

Given a partial solution $(T_1, \mathcal{S}_1, p_1)$ as input, the point completion layer algorithm proceeds as follows. First, the algorithm initializes an empty hash table Z in preparation for isomorph rejection. If $(T_1, \mathcal{S}_1, p_1)$ is complete, then the algorithm reports $(T_1 \cup \{p_1\}, \mathcal{S}_1)$ to the transversal layer—also indicating that p_1 was the last completed point. Next, the algorithm loops through all orbits $\mathcal{O} \in H \backslash \Omega$ that satisfy $T_1 \downarrow \mathcal{O} = \emptyset$ and $\{p_1\} \downarrow \mathcal{O} = \mathcal{O}$. For every such orbit $\mathcal{O}$, the algorithm performs isomorph rejection on the extension $(T, \mathcal{S}, p) \leftarrow (T_1, \mathcal{S}_1 \cup \mathcal{O}, p_1)$. If $(T, \mathcal{S}, p)$ is accepted, then the point completion layer is recursively invoked with input $(T, \mathcal{S}, p)$; otherwise $(T, \mathcal{S}, p)$ is rejected. After all orbits $\mathcal{O}$ have been considered, the algorithm releases the hash table Z and returns.

4.1.2. Isomorph rejection. The isomorph rejection step consists of two tests, both of which must pass for $(T, \mathcal{S}, p)$ to be accepted. The first test distinguishes an extending H-orbit $\mathcal{O}_0$ from $(T, \mathcal{S}, p)$ and rejects $(T, \mathcal{S}, p)$ unless $\mathcal{O}_0 \cong_{\mathrm{Aut}_{N_G(H)}(T,\mathcal{S},p)} \mathcal{O}$, where $\mathcal{O}$ is the H-orbit most recently added to $\mathcal{S}$. The distinguishing operation is canonical in the following sense:

(4.2) If $\mathcal{O}_0$ and $\mathcal{O}_0'$ are the extending orbits distinguished from $(T, \mathcal{S}, p)$ and $(T', \mathcal{S}', p')$, respectively, and $(T, \mathcal{S}, p) \cong_{N_G(H)} (T', \mathcal{S}', p')$, then $(T, \mathcal{S}, p, \mathcal{O}_0) \cong_{N_G(H)} (T', \mathcal{S}', p', \mathcal{O}_0')$.

Example 11. In our applications, the distinguishing operation is implemented as follows. Put

$$\begin{aligned}
\mathcal{R} &\leftarrow \mathcal{S} \cup \{T\} \cup \{\{p\}\} \cup \{\{p\}\},\\
\hat{\mathcal{S}} &\leftarrow \kappa(\mathcal{R})\mathcal{S},\\
\hat{T} &\leftarrow \kappa(\mathcal{R})T,\\
\hat{p} &\leftarrow \kappa(\mathcal{R})p,\\
\hat{B} &\leftarrow \min(\hat{\mathcal{S}} - (\hat{T} \downarrow \hat{\mathcal{S}})),\\
\mathcal{O}_0 &\leftarrow \kappa(\mathcal{R})^{-1}\{h\hat{B} : h \in H\}.
\end{aligned} \tag{4.3}$$

A few remarks are in order. First, we assume that the $N_G(H)$-canonical labeling map κ can handle set systems with repeated blocks. Second, we assume that the three-tuple $(T, \mathcal{S}, p)$ can be reconstructed up to $N_G(H)$-isomorphism from $\mathcal{R}$, and that $\mathrm{Aut}_{N_G(H)}(T, \mathcal{S}, p) = \mathrm{Aut}_{N_G(H)}(\mathcal{R})$. This is the case in our applications because the point p is the only point in $\mathcal{R}$ which appears in two singleton blocks, and the points in T are the only points (with the possible exception of p) that are incident to $r + 1$ blocks in $\mathcal{R}$.

The second isomorph rejection test computes an $N_G(H)$-canonical representative for $(T, \mathcal{S}, p)$. The test rejects $(T, \mathcal{S}, p)$ if the canonical representative occurs in the hash table Z; otherwise the test accepts $(T, \mathcal{S}, p)$, and the canonical representative is inserted into Z.

Example 12. With the assumptions of Example 11, we can use $(\hat{T}, \hat{\mathcal{S}}, \hat{p})$ computed in (4.3) as the $N_G(H)$-canonical representative.

4.1.3. Correctness. We proceed to show that the point completion layer is correct; that is, it produces exactly one extension from every $N_G(H)$-isomorphism class of complete extensions of $(T_0, \mathcal{S}_0, p_0)$.

LEMMA 4.1. *Let $(T, \mathcal{S}, p)$ be constructed from $(T_1, \mathcal{S}_1, p_1)$ by adding the H-orbit $\mathcal{O}$. Let $(T', \mathcal{S}', p')$ be constructed from $(T_1', \mathcal{S}_1', p_1')$ by adding the H-orbit $\mathcal{O}'$. Let $\mathcal{O}_0$ and $\mathcal{O}_0'$ be the extending orbits distinguished from $(T, \mathcal{S}, p)$ and $(T', \mathcal{S}', p')$, respectively. If $(T, \mathcal{S}, p) \cong_{N_G(H)} (T', \mathcal{S}', p')$, $\mathcal{O} \cong_{\mathrm{Aut}_{N_G(H)}(T,\mathcal{S},p)} \mathcal{O}_0$, and $\mathcal{O}' \cong_{\mathrm{Aut}_{N_G(H)}(T',\mathcal{S}',p')} \mathcal{O}_0'$, then $(T_1, \mathcal{S}_1, p_1) \cong_{N_G(H)} (T_1', \mathcal{S}_1', p_1')$.*

Proof. By (4.2) and $(T, \mathcal{S}, p) \cong_{N_G(H)} (T', \mathcal{S}', p')$ we obtain $(T, \mathcal{S}, p, \mathcal{O}_0) \cong_{N_G(H)} (T', \mathcal{S}', p', \mathcal{O}_0')$. Thus, it follows from $\mathcal{O} \cong_{\mathrm{Aut}_{N_G(H)}(T,\mathcal{S},p)} \mathcal{O}_0$ and $\mathcal{O}' \cong_{\mathrm{Aut}_{N_G(H)}(T',\mathcal{S}',p')} \mathcal{O}_0'$ that $(T, \mathcal{S}, p, \mathcal{O}) \cong_{N_G(H)} (T', \mathcal{S}', p', \mathcal{O}')$. The claim follows because $(T, \mathcal{S} - \mathcal{O}, p) = (T_1, \mathcal{S}_1, p_1)$ and $(T', \mathcal{S}' - \mathcal{O}', p') = (T_1', \mathcal{S}_1', p_1')$. □

THEOREM 4.2. *Let $(T', \mathcal{S}', p')$ be an extension of $(T_0, \mathcal{S}_0, p_0)$. Then, there exists a unique extension $(T, \mathcal{S}, p)$ of $(T_0, \mathcal{S}_0, p_0)$ such that $(T, \mathcal{S}, p) \cong_{N_G(H)} (T', \mathcal{S}', p')$ and the point completion layer is invoked with input $(T, \mathcal{S}, p)$ in the search for extensions of $(T_0, \mathcal{S}_0, p_0)$.*

Proof. Let m be the level of $(T', \mathcal{S}', p')$. We proceed by induction on m. The base case $m = 0$ is obvious. Suppose the claim holds for all extensions $(T', \mathcal{S}', p')$ on level m. Lemma 4.1 and the structure of the isomorph rejection tests imply that the point completion layer is invoked at most once for every $N_G(H)$-isomorphism class of extensions on level $m + 1$.

Let $(T', \mathcal{S}', p')$ be an arbitrary extension of $(T_0, \mathcal{S}_0, p_0)$ on level $m+1$. To complete the inductive step, it suffices to show that the point completion layer is invoked for an extension $N_G(H)$-isomorphic to $(T', \mathcal{S}', p')$. Let $\mathcal{O}_0'$ be the extending orbit distinguished from $(T', \mathcal{S}', p')$. Clearly, $(T', \mathcal{S}' - \mathcal{O}_0', p')$ is an extension on level m. By the inductive hypothesis, there exists a unique extension $(T_1, \mathcal{S}_1, p_1)$ such that $(T', \mathcal{S}' - \mathcal{O}_0', p') \cong_{N_G(H)} (T_1, \mathcal{S}_1, p_1)$ and the point completion layer is invoked with input $(T_1, \mathcal{S}_1, p_1)$. Let $g' \in N_G(H)$ take $(T', \mathcal{S}' - \mathcal{O}_0', p')$ to $(T_1, \mathcal{S}_1, p_1)$. Put $\mathcal{O} = g'\mathcal{O}_0'$ and $(T, \mathcal{S}, p) = (T_1, \mathcal{S}_1 \cup \mathcal{O}, p_1)$. By the structure of the algorithm, the orbit $\mathcal{O}$ is considered during the point completion layer invocation with input $(T_1, \mathcal{S}_1, p_1)$ and $(T, \mathcal{S}, p)$ is constructed.

We proceed to show that $(T, \mathcal{S}, p)$ passes the first isomorph rejection test. Let $\mathcal{O}_0$ be the extending orbit distinguished from $(T, \mathcal{S}, p)$. Because $(T, \mathcal{S}, p) \cong_{N_G(H)} (T', \mathcal{S}', p')$, we have $(T, \mathcal{S}, p, \mathcal{O}_0) \cong_{N_G(H)} (T', \mathcal{S}', p', \mathcal{O}_0')$ by (4.2). Let $g \in N_G(H)$ take $(T', \mathcal{S}', p', \mathcal{O}_0')$ to $(T, \mathcal{S}, p, \mathcal{O}_0)$. Now $g'g^{-1} \in \mathrm{Aut}_{N_G(H)}(T, \mathcal{S}, p)$ takes $\mathcal{O}_0$ to $\mathcal{O}$. Thus, $\mathcal{O}_0 \cong_{\mathrm{Aut}_{N_G(H)}(T,\mathcal{S},p)} \mathcal{O}$, so $(T, \mathcal{S}, p)$ passes the first isomorph rejection test. Then

$(T, \mathcal{S}, p)$ either passes the second isomorph rejection test (and the point completion layer is invoked with input $(T, \mathcal{S}, p)$) or the second test rejects $(T, \mathcal{S}, p)$. In the latter case the canonical representative of $(T, \mathcal{S}, p)$ occurs in the hash table Z, which implies that the point completion layer has already been invoked with an extension $N_G(H)$-isomorphic to $(T', \mathcal{S}', p')$. □

4.2. The transversal layer. Given a group $H \in \mathcal{H}$ as input, the transversal layer generates the H-seeds by completing one point at a time in a backtrack search.

An ℓ-subset $T \subseteq \Pi$ is an *ℓ-transversal* for $\mathcal{S} \subseteq_H \Omega$ if $(T, \mathcal{S})$ has property $P^{\downarrow}$ and $T \downarrow \mathcal{S} = \mathcal{S}$.

The transversal layer selects points to be completed by means of transversal invariants. The same invariants are then used to distinguish the set T_0 in constructing a parent seed, which ensures that every parent seed $(H_0, T_0, \mathcal{S}_0)$ with H_0 conjugate to H is isomorphic to an H-seed.

A *transversal invariant* for H is a function I_ℓ that associates to every $\mathcal{S} \subseteq_H \Omega$ a (possibly empty) set $I_\ell(\mathcal{S})$ of ℓ-transversals for $\mathcal{S}$ such that

$$gI_\ell(\mathcal{S}) = I_\ell(g\mathcal{S}) \text{ for all } g \in N_G(H). \tag{4.4}$$

A transversal invariant for H can be extended to any conjugate H_0 of H and $\mathcal{S}_0 \subseteq_{H_0} \Omega$ by defining

$$I_\ell(H_0, \mathcal{S}_0) = gI_\ell(g^{-1}\mathcal{S}_0), \tag{4.5}$$

where $g \in G$ is any group element that satisfies $gHg^{-1} = H_0$. It follows from (4.4) that this extension is well defined. Furthermore, the extended transversal invariant satisfies

$$gI_\ell(H, \mathcal{S}) = I_\ell(gHg^{-1}, g\mathcal{S}) \text{ for all } g \in G. \tag{4.6}$$

Here are some examples of tranversal invariants.

Example 13. For a nonnegative integer f, let $I_\ell^{\mathrm{fix}(f)}(\mathcal{S})$ consist of all ℓ-transversals T for $\mathcal{S}$ such that $|T \cap \mathrm{fix}(H)| = \min(\ell, f)$.

Example 14. Let $I_\ell^{\mathrm{blk}}(\mathcal{S})$ consist of all ℓ-transversals T for $\mathcal{S}$ such that there exists a block $B \in \mathcal{S}$ with $T \subseteq B$.

Example 15. Let $I_\ell^{\mathrm{orb}}(\mathcal{S})$ consist of all ℓ-transversals T for $\mathcal{S}$ such that no two points in T are in the same H-orbit.

LEMMA 4.3. *Let I'_ℓ and I''_ℓ be transversal invariants. Then, the intersection $I_\ell(\mathcal{S}) = I'_\ell(\mathcal{S}) \cap I''_\ell(\mathcal{S})$ is a transversal invariant.*

4.2.1. Partial solutions. We proceed to describe the partial solutions in the transversal layer backtrack search. Let k be a nonnegative integer and let $I_0, I_1, \ldots, I_k$ be a sequence of transversal invariants. We denote by $\mathscr{S}_\ell$ the set of all pairs $(T, \mathcal{S})$ that satisfy $T \in I_\ell(\mathcal{S})$. A partial solution in the transversal layer is a pair $(T, \mathcal{S}) \in \mathscr{S}_\ell$. The *level* of a partial solution $(T, \mathcal{S}) \in \mathscr{S}_\ell$ is ℓ. A partial solution $(T, \mathcal{S})$ is *complete*—that is, $(T, \mathcal{S})$ is an H-seed—if $\ell = k$.

Before describing the algorithm, we must still connect the partial solutions on successive levels so that the search can generate all $N_G(H)$-nonisomorphic partial solutions on level $\ell + 1$ from the $N_G(H)$-nonisomorphic solutions on level ℓ. We establish this connection between successive levels via "parent points" and "extending points" as follows. (Note that the existence of these point sets for a given sequence of invariants $I_0, I_1, \ldots, I_k$ obviously depends on whether the invariants are sufficiently "compatible" with each other to admit such sets. Furthermore, note that the existence also depends on the hereditary property (4.1) for $P^{\downarrow}$ because any $(T,\mathcal{S}) \in \mathscr{S}_\ell$ must have property $P^{\downarrow}$.)

For $1 \le \ell \le k$, we associate with every $(T,\mathcal{S}) \in \mathscr{S}_\ell$ a nonempty *parent point set* $\mathrm{par}_\ell(T,\mathcal{S}) \subseteq T$ such that

$$g \cdot \mathrm{par}_\ell(T,\mathcal{S}) = \mathrm{par}_\ell(gT, g\mathcal{S}) \qquad \text{for all } g \in N_G(H), \text{ and} \tag{4.7}$$

$$(T - \{p\}, (T - \{p\}){\downarrow}\mathcal{S}) \in \mathscr{S}_{\ell-1} \quad \text{for all } p \in \mathrm{par}_\ell(T,\mathcal{S}). \tag{4.8}$$

For $0 \le \ell \le k-1$, we associate with every $(T,\mathcal{S}) \in \mathscr{S}_\ell$ an *extending point set* $\mathrm{ext}_\ell(T,\mathcal{S}) \subseteq \Pi - T$ such that

$$g \cdot \mathrm{ext}_\ell(T,\mathcal{S}) = \mathrm{ext}_\ell(gT, g\mathcal{S}) \text{ for all } g \in N_G(H), \text{ and} \tag{4.9}$$

$$\begin{aligned} &p \in \mathrm{ext}_\ell(T,\mathcal{S}) \text{ for all } (T',\mathcal{S}') \in \mathscr{S}_{\ell+1} \text{ and all} \\ &\quad p \in \mathrm{par}_{\ell+1}(T',\mathcal{S}') \text{ such that } (T,\mathcal{S}) = (T' - \{p\}, (T' - \{p\}){\downarrow}\mathcal{S}'). \end{aligned} \tag{4.10}$$

Example 16. Consider the sequence $I_0^{\mathrm{fix}(f)}, I_1^{\mathrm{fix}(f)}, \ldots, I_k^{\mathrm{fix}(f)}$ of transversal invariants. For $1 \le \ell \le k$ and $(T,\mathcal{S}) \in \mathscr{S}_\ell$, define

$$\mathrm{par}_\ell(T,\mathcal{S}) = \begin{cases} T \cap \mathrm{fix}(H) & \text{if } \ell \le f, \text{ and} \\ T - \mathrm{fix}(H) & \text{if } \ell > f. \end{cases}$$

For $0 \le \ell \le k-1$ and $(T,\mathcal{S}) \in \mathscr{S}_\ell$, define

$$\mathrm{ext}_\ell(T,\mathcal{S}) = \begin{cases} \mathrm{fix}(H) - T & \text{if } \ell < f, \text{ and} \\ \Pi - (T \cup \mathrm{fix}(H)) & \text{if } \ell \ge f. \end{cases}$$

Example 17. Consider the sequence $I_0^{\mathrm{blk}}, I_1^{\mathrm{blk}}, \ldots, I_k^{\mathrm{blk}}$ of transversal invariants. For $1 \le \ell \le k$ and $(T,\mathcal{S}) \in \mathscr{S}_\ell$, define $\mathrm{par}_\ell(T,\mathcal{S}) = T$. For $1 \le \ell \le k-1$ and $(T,\mathcal{S}) \in \mathscr{S}_\ell$, let $\mathrm{ext}_\ell(T,\mathcal{S})$ be the set of all points $p \in \Pi - T$ such that there exists a block $B \in \mathcal{S}$ with $p \in B$ and $T \subseteq B$. For $\ell = 0$, put $\mathrm{ext}_\ell(T,\mathcal{S}) = \Pi$.

Example 18. Consider the sequence $I_0^{\mathrm{orb}}, I_1^{\mathrm{orb}}, \ldots, I_k^{\mathrm{orb}}$ of transversal invariants. For $1 \le \ell \le k$ and $(T,\mathcal{S}) \in \mathscr{S}_\ell$, define $\mathrm{par}_\ell(T,\mathcal{S}) = T$. For $0 \le \ell \le k-1$ and $(T,\mathcal{S}) \in \mathscr{S}_\ell$, let $\mathrm{ext}_\ell(T,\mathcal{S})$ be the set of all points $p \in \Pi$ such that, for all $q \in T$, p and q occur on different H-orbits in Π.

Example 19. Any intersection of the transversal invariants on Examples 13, 14, and 15 defines a transversal invariant by Lemma 4.3. For such an invariant, the associated parent point and extending point sets are obtained as the intersection of the corresponding sets in Examples 16, 17, and 18.

4.2.2. Algorithm. We are now ready to describe the transversal layer algorithm. We assume that a sequence $I_0, I_1, \ldots, I_k$ of transversal invariants has been selected for H. Initially, the algorithm is invoked with $T_1 = \emptyset$, $\mathcal{S}_1 = \emptyset$.

Let $(T_1, \mathcal{S}_1) \in \mathscr{S}_\ell$ be given as input to the algorithm. If $\ell = k$, the algorithm outputs $(T_1, \mathcal{S}_1)$ as an H-seed and returns. Otherwise, the algorithm initializes an empty hash table Y in preparation for isomorph rejection. Next, the algorithm loops through all points $p \in \mathrm{ext}_\ell(T_1, \mathcal{S}_1)$. For each such point p, the algorithm invokes the point completion layer with input $(T_1, \mathcal{S}_1, p)$. For every pair $(T,\mathcal{S})$ reported by the point completion layer, the algorithm performs an isomorph rejection step. If $(T,\mathcal{S})$ passes the isomorph rejection step, then the transversal layer is recursively invoked with input $(T,\mathcal{S})$; otherwise $(T,\mathcal{S})$ is rejected. After all points p have been considered, the algorithm releases the hash table Y and returns.

4.2.3. Isomorph rejection. The isomorph rejection step consists of two tests, both of which must pass for $(T, \mathcal{S})$ to be accepted. Let $p \in T$ be the most recently completed point in $(T, \mathcal{S})$. The first test starts by distinguishing a parent point $p_0 \in \mathrm{par}_\ell(T, \mathcal{S})$, where the distinguishing operation is canonical in the following sense:

(4.11) If p_0 and p_0' are the parent points distinguished from $(T, \mathcal{S})$ and $(T', \mathcal{S}')$, respectively, and $(T, \mathcal{S}) \cong_{N_G(H)} (T', \mathcal{S}')$, then $(T, \mathcal{S}, p_0) \cong_{N_G(H)} (T', \mathcal{S}', p_0')$.

The first test accepts $(T, \mathcal{S})$ if and only if $p_0 \cong_{\mathrm{Aut}_{N_G(H)}(T,\mathcal{S})} p$.

Example 20. In our applications, the distinguishing operation is implemented as follows. Put

$$\begin{aligned} \mathcal{R} &\leftarrow \mathcal{S} \cup \{T\}, \\ \hat{\mathcal{S}} &\leftarrow \kappa(\mathcal{R})\mathcal{S}, \\ \hat{T} &\leftarrow \kappa(\mathcal{R})T, \\ \hat{p} &\leftarrow \min \mathrm{par}_{|\hat{T}|}(\hat{T}, \hat{\mathcal{S}}), \\ p_0 &\leftarrow \kappa(\mathcal{R})^{-1}\hat{p}. \end{aligned} \tag{4.12}$$

Here we assume again that the $N_G(H)$-canonical labeling map κ can handle set systems with repeated blocks; cf. Example 11. Second, we assume that $(T, \mathcal{S})$ can be reconstructed up to $N_G(H)$-isomorphism from $\mathcal{R}$ and that $\mathrm{Aut}_{N_G(H)}(T, \mathcal{S}) = \mathrm{Aut}_{N_G(H)}(\mathcal{R})$. This is the case in our applications because the points in T are the only points that are incident to $r + 1$ blocks in $\mathcal{R}$.

The second isomorph rejection test computes an $N_G(H)$-canonical representative for $(T, \mathcal{S})$. The test rejects $(T, \mathcal{S})$ if the canonical representative occurs in the hash table Y; otherwise the test accepts $(T, \mathcal{S})$ and the canonical representative is inserted into Y.

Example 21. With the assumptions of Example 20, $(\hat{T}, \hat{\mathcal{S}})$ computed in (4.12) can be used as the $N_G(H)$-canonical representative of $(T, \mathcal{S})$.

4.2.4. Correctness. We proceed to prove that the transversal layer is invoked with exactly one pair $(T, \mathcal{S})$ from every $N_G(H)$-isomorphism class in $\mathscr{S}_\ell$ for all $0 \leq \ell \leq k$. We say that $(T, \mathcal{S}) \in \mathscr{S}_{\ell+1}$ *extends* $(T_1, \mathcal{S}_1) \in \mathscr{S}_\ell$ by completing $p \in T$ if $(T - \{p\}, (T - \{p\}) \downarrow \mathcal{S}) \cong_{N_G(H)} (T_1, \mathcal{S}_1)$.

LEMMA 4.4. *Let $(T, \mathcal{S})$ extend $(T_1, \mathcal{S}_1)$ by completing p, and let p_0 be the parent point distinguished from $(T, \mathcal{S})$. Similarly, let $(T', \mathcal{S}')$ extend $(T_1', \mathcal{S}_1')$ by completing p', and let p_0' be the parent point distinguished from $(T', \mathcal{S}')$. If $(T, \mathcal{S}) \cong_{N_G(H)} (T', \mathcal{S}')$, $p_0 \cong_{\mathrm{Aut}_{N_G(H)}(T,\mathcal{S})} p$, and $p_0' \cong_{\mathrm{Aut}_{N_G(H)}(T',\mathcal{S}')} p'$, then $(T_1, \mathcal{S}_1) \cong_{N_G(H)} (T_1', \mathcal{S}_1')$.*

Proof. We proceed along the following sequence of isomorphisms:

$$\begin{aligned} (T_1, \mathcal{S}_1) &\cong_{N_G(H)} (T - \{p\}, (T - \{p\}) \downarrow \mathcal{S}) \\ &\cong_{N_G(H)} (T - \{p_0\}, (T - \{p_0\}) \downarrow \mathcal{S}) \\ &\cong_{N_G(H)} (T' - \{p_0'\}, (T' - \{p_0'\}) \downarrow \mathcal{S}') \\ &\cong_{N_G(H)} (T' - \{p'\}, (T' - \{p'\}) \downarrow \mathcal{S}') \\ &\cong_{N_G(H)} (T_1', \mathcal{S}_1'). \end{aligned}$$

The first and last isomorphism relations hold by definition. The second isomorphism relation holds because $p_0 \cong_{\mathrm{Aut}_{N_G(H)}(T,\mathcal{S})} p$. The third relation holds because

$(T,\mathcal{S}) \cong_{N_G(H)} (T',\mathcal{S}')$ implies $(T,\mathcal{S},p_0) \cong_{N_G(H)} (T',\mathcal{S}',p_0')$ by (4.11). The fourth relation holds because $p_0' \cong_{\mathrm{Aut}_{N_G(H)}(T',\mathcal{S}')} p'$. □

THEOREM 4.5. *Let $0 \le \ell \le k$ and let $(T',\mathcal{S}') \in \mathscr{S}_\ell$. Then there exists a unique $(T,\mathcal{S}) \in \mathscr{S}_\ell$ such that $(T,\mathcal{S}) \cong_{N_G(H)} (T',\mathcal{S}')$ and the transversal layer is invoked with input $(T,\mathcal{S})$.*

Proof. By induction on ℓ. The base case $\ell = 0$ is obvious. Suppose the claim holds for all $(T',\mathcal{S}') \in \mathscr{S}_\ell$. Lemma 4.4 and the structure of the isomorph rejection tests imply that the transversal layer is invoked at most once for every $N_G(H)$-isomorphism class in $\mathscr{S}_{\ell+1}$.

Let $(T',\mathcal{S}') \in \mathscr{S}_{\ell+1}$. Let $p_0' \in \mathrm{par}_{\ell+1}(T',\mathcal{S}')$ be the parent point distinguished from $(T',\mathcal{S}')$. By (4.8), $(T'-\{p_0'\},(T'-\{p_0'\})\downarrow\mathcal{S}') \in \mathscr{S}_\ell$. By the induction hypothesis, there exists a unique $(T_1,\mathcal{S}_1) \in \mathscr{S}_\ell$ such that the transversal layer is invoked with $(T_1,\mathcal{S}_1)$ and $(T_1,\mathcal{S}_1) \cong_{N_G(H)} (T'-\{p_0'\},(T'-\{p_0'\})\downarrow\mathcal{S}')$. Let $g' \in N_G(H)$ take $(T'-\{p_0'\},(T'-\{p_0'\})\downarrow\mathcal{S}')$ to $(T_1,\mathcal{S}_1)$. By (4.10) and (4.9), $g'p_0' \in \mathrm{ext}_\ell(T_1,\mathcal{S}_1)$. Thus, the point completion layer is invoked with input $(T_1,\mathcal{S}_1,g'p_0')$ during the transversal layer invocation with input $(T_1,\mathcal{S}_1)$. Because $(T'-\{p_0'\},\mathcal{S}',p_0')$ is a complete extension of $(T'-\{p_0'\},(T'-\{p_0'\})\downarrow\mathcal{S}',p_0')$, it follows that $(g'T'-\{g'p_0'\},g'\mathcal{S}',g'p_0')$ is a complete extension of $(T_1,\mathcal{S}_1,g'p_0')$.

By Theorem 4.2, the point completion layer reports to the transversal layer exactly one pair $(T,\mathcal{S})$ such that $(T,\mathcal{S},p) \cong_{N_G(H)} (g'T',g'\mathcal{S}',g'p_0')$, where p is the most recently completed point in $(T,\mathcal{S})$. We show that $(T,\mathcal{S})$ is accepted in the first isomorph rejection test. Let $g \in N_G(H)$ take $(T,\mathcal{S},p)$ to $(g'T',g'\mathcal{S}',g'p_0')$. Let $p_0 \in \mathrm{par}_{\ell+1}(T,\mathcal{S})$ be the parent point distinguished from $(T,\mathcal{S})$. Because $(T,\mathcal{S}) \cong_{N_G(H)} (T',\mathcal{S}')$, we have $(T,\mathcal{S},p_0) \cong_{N_G(H)} (T',\mathcal{S}',p_0')$ by (4.11). Let $g_0 \in N_G(H)$ take $(T,\mathcal{S},p_0)$ to $(T',\mathcal{S}',p_0')$. Now, $g^{-1}g'g_0 \in \mathrm{Aut}_{N_G(H)}(T,\mathcal{S})$ shows that $p_0 \cong_{\mathrm{Aut}_{N_G(H)}(T,\mathcal{S})} p$. Thus, $(T,\mathcal{S})$ is accepted in the first isomorph rejection test, which implies—by the structure of the second test—that the transversal layer is invoked with a pair $N_G(H)$-isomorphic to $(T',\mathcal{S}')$. □

4.2.5. Parent seeds and *H*-seeds. It remains to specify in detail the distinguishing operation for the set T_0 that was left unspecified in section 3.1.

Let $I_0, I_1, \ldots, I_k$ be the sequence of transversal invariants used to construct the H-seeds. Extend the transversal invariant I_k to conjugates of H by (4.5).

Let $\mathscr{X}$ be a target system for which the distinguished subgroup $H_0 \le \mathrm{Aut}(\mathscr{X})$ is conjugate to H. In distinguishing the set T_0 based on $\mathscr{X}$ and H_0, we now require that $T_0 \in I_k(H_0,(H_0,T_0)\downarrow\mathscr{X})$. (Whether such a set T_0 actually exists obviously depends on the choice of I_k. We assume that I_k has been selected so that such a T_0 exists.)

Example 22. Example 3 illustrates the resulting distinguishing procedure for T_0 in the case of the transversal invariant $I_k^{\mathrm{fix}(f)}$. If we add the requirement (3.5) to (3.4), then the distinguishing procedure corresponds to the transversal invariant $I_k^{\mathrm{fix}(f)} \cap I_k^{\mathrm{blk}}$. Analogously, (3.4), (3.5), and (3.6) together correspond to $I_k^{\mathrm{fix}(f)} \cap I_k^{\mathrm{blk}} \cap I_k^{\mathrm{orb}}$.

From $T_0 \in I_k(H_0,(H_0,T_0)\downarrow\mathscr{X})$ it immediately follows that the resulting parent seed $(H_0,T_0,\mathcal{S}_0)$, where $\mathcal{S}_0 = (H_0,T_0)\downarrow\mathscr{X}$, is G-isomorphic to an H-seed. To see this, let $g \in G$ satisfy $gH_0g^{-1} = H$, and observe that $(gT_0,g\mathcal{S}_0) \in \mathscr{S}_k$. By Theorem 4.5, the seed generation algorithm generates an H-seed isomorphic to $(H,gT_0,g\mathcal{S}_0)$. Repeating this argument for all $H \in \mathcal{H}$, we obtain that the seed generation algorithm satisfies (3.8) for all $H \in \mathcal{H}$, which implies correctness of the classification algorithm by Theorems 3.1 and 3.2.

5. Performance and implementation details. To make the classification algorithm practical, we require fast implementations of many of the primitives employed, such as the canonical labeling maps. This section discusses implementation of the required nontrivial primitives to enable reproducibility of the classification result for Steiner triple systems of order 21 and to expose some of the performance bottlenecks in the framework.

5.1. Canonical labeling for target systems. Excluding the extension phase from seeds to target systems, the most performance-critical part of the algorithm is the G-canonical labeling map that must be evaluated for every target system occurring in the search. For this task we apply the graph canonical labeling package *nauty* [53, 54], which is well tested and exhibits good practical performance. Furthermore, *nauty* also computes a set of generators for $\mathrm{Aut}(\mathcal{X})$ as a side effect of computing $\kappa(\mathcal{X})$, both of which are required in the isomorph rejection phase.

To apply *nauty*, we must encode a set system as a graph. (Here we assume that G is the symmetric group on Π or the stabilizer of a partition of Π in $\mathrm{Sym}(\Pi)$. For other permutation groups, we must also encode the group into the input graph for *nauty*, which may not be straightforward.) Two standard graph transformations of set systems into graphs are the incidence graph [28, Remark 9.41] and the line graph [1]; the latter encoding is often more efficient but not always applicable because a set system is not always (strongly) reconstructible from its line graph.

Example 23. For an $\mathrm{STS}(v)$, the line graph transformation is applicable for $v \geq 19$ [1, 20, 21].

Invariants can be employed to speed up the operation of *nauty* on a target system $\mathcal{X}$.

Example 24. For an $\mathrm{STS}(v)$, we take advantage of *Pasch configurations* or *quadrilaterals*, which are sets of four blocks of the form

$$\{u,v,w\}, \quad \{u,x,y\}, \quad \{v,x,z\}, \quad \{w,y,z\}. \tag{5.1}$$

For every block $B \in \mathcal{X}$, we compute the number $P(B)$ of Pasch configurations in $\mathcal{X}$ in which B occurs, and use the values $P(B)$ to construct an ordered partition of the blocks. Two blocks are in the same cell of the partition if and only if they have equal $P(B)$-values, and the cells are ordered by their $P(B)$-values. This ordered partition is then input to *nauty* along with the line graph of $\mathcal{X}$ to speed up the computation; cf. [31]. Algorithms for finding the Pasch configurations in an $\mathrm{STS}(v)$ are considered in [68].

To obtain further speedup, it is possible to use invariants more extensively in the isomorph rejection phase to avoid an expensive full canonical labeling computation whenever possible; see [7, 31, 32, 55] for examples of the use of invariants. However, incorporating such techniques to the present framework remains a topic of future research. (One possibility is to reverse the order in which H_0, T_0 are distinguished in constructing the parent seed. In this case we can use invariants in distinguishing T_0 from $\mathcal{X}$, which should enable a fast rejection strategy analogous to the ones employed in [31, 32]. On the downside, it appears that constraining the structure of the seeds becomes more difficult in this setting.)

5.2. Permutation group algorithms. In our applications all the groups manipulated by the classification algorithm are permutation groups on Π, which are represented using a base and a strong generating set; see [9, 66].

The tasks that are not performance-critical—such as constructing H-orbits on Ω and generators for the normalizer $N_G(H)$—are solved in a preprocessing stage using dedicated software [27].

The performance-critical parts in the top-level algorithm involving computation with permutation groups—that is, distinguishing the subgroup H_0 in a parent seed and testing whether $(H, T, \mathcal{S}) \cong_{\mathrm{Aut}(\mathcal{X})} (H_0, T_0, \mathcal{S}_0)$—are at present implemented by exhaustive search on $\mathrm{Aut}(\mathcal{X})$; see Examples 2 and 7. This is sufficient in our applications, where most target systems have a small automorphism group and the groups in $\mathcal{H}$ have small prime order. For larger groups, both the distinguishing operation and the isomorphism test require nontrivial algorithms (see [47, 48, 66]) and present a possible performance bottleneck.

The performance-critical isomorph rejection tests in the seed generation algorithm can be implemented with orbit computations (see [66, sect. 2.1]) once generators for $\mathrm{Aut}_{N_G(H)}(T, \mathcal{S}, p)$ (point completion layer) and $\mathrm{Aut}_{N_G(H)}(T, \mathcal{S})$ (transversal layer) are available. Both generator sets are obtained as a side effect of evaluating the $N_G(H)$-canonical labeling map for the set system $\mathcal{R}$ derived from $(T, \mathcal{S}, p)$ and $(T, \mathcal{S})$, respectively; see Examples 11 and 20.

5.3. Computing $N_G(H)$-canonical labeling for set systems. The central primitive in the seed generation algorithm is a canonical labeling map that takes a set system $\mathcal{X} \subseteq \Omega$ and a permutation group $K \leq \mathrm{Sym}(\Pi)$ as input and computes a K-canonical labeling for $\mathcal{X}$. Furthermore, we also require a set of generators for $\mathrm{Aut}_K(\mathcal{X})$.

Such a primitive can be implemented by combining the partition refinement approach in [53] with backtrack search along a chain of point stabilizer subgroups in K. This is essentially an application of the partition method developed in [47, 48]; however, the property that we want to compute here is slightly different from the subgroup-type and coset-type properties treated in [48]. Namely, we want to solve both a subgroup-type problem—that is, to determine generators for $\mathrm{Aut}_K(\mathcal{X})$—and a "canonical coset"-type problem to determine $\kappa(\mathcal{X})$. (By "canonical" we mean here that $\kappa(g\mathcal{X})g\mathcal{X} = \kappa(\mathcal{X})\mathcal{X}$ must hold for all $g \in K$ and all $\mathcal{X} \subseteq \Omega$.) Consequently, our implementation of the K-canonical labeling map follows the ideas in [53] to a large extent. The main differences are that we work along a point stabilizer chain in K and—to keep the implementation simple—do not implement many of the clever pruning techniques in [53]. A further simplification is obtained by working with a fixed base for K, although dynamic base change is certainly possible; cf. [8, 10, 16, 66]. To achieve a better degree of reproducibility for the classification results, we sketch the structure of the implementation.

We require some standard definitions related to ordered partitions. For simplicity, let $\Pi = \{1, 2, \ldots, n\}$. An *ordered partition* of Π is a sequence $W = (W_1, W_2, \ldots, W_s)$ of subsets of Π—called *cells*—such that $\{W_1, W_2, \ldots, W_s\}$ is a partition of Π. For $p \in \Pi$, we write $W(p)$ for the index of the cell in which p occurs; for example, if $p \in W_j$, then $W(p) = j$. Let $W = (W_1, W_2, \ldots, W_s)$ and $Z = (Z_1, Z_2, \ldots, Z_t)$ be ordered partitions of Π. The *intersection* $W \wedge Z$ is the ordered partition of Π whose cells are the nonempty sets of the form $W_i \cap Z_j$. The cells in $W \wedge Z$ are ordered so that $W_{i_1} \cap Z_{j_1}$ precedes $W_{i_2} \cap Z_{j_2}$ if and only if either $i_1 < i_2$ or both $i_1 = i_2$ and $j_1 < j_2$. (Note that the intersection operation is noncommutative.)

The canonical labeling algorithm relies extensively on the following refinement procedure for ordered partitions, which is analogous to the basic equitable refinement procedure in [53]. Given a set system $\mathcal{X}$ and an ordered partition W of Π as input, we iterate the following steps until the partition W no longer changes. First, for every $p \in \Pi$, we compute the multiset $m(p) = \{\{W(q) : q \in B\} : p \in B \in \mathcal{X}\}$ consisting of multisets of positive integers. We then form an ordered partition Z of Π by placing two points $p_1, p_2 \in \Pi$ in the same cell if and only if $m(p_1) = m(p_2)$; the cells are ordered based on some (say, lexicographic) order on the multisets $m(p)$. Finally, we complete the iteration by setting $W \leftarrow W \wedge Z$. We denote the ordered partition resulting from this procedure by $e(\mathcal{X}, W)$. By the structure of the procedure,

$$(5.2) \qquad g \cdot e(\mathcal{X}, W) = e(g\mathcal{X}, gW) \quad \text{for all } g \in \mathrm{Sym}(\Pi),$$

where g acts on an ordered partition of Π by permuting the points in the cells; the ordering of the cells does not change.

We now sketch the canonical labeling algorithm. Let $K \leq \mathrm{Sym}(\Pi)$ and $\mathcal{X} \subseteq \Omega$ be given as input to the algorithm. We work with the stabilizer chain

$$K = K_1 \geq K_2 \geq K_3 \geq \cdots \geq K_{n+1} = \{1\},$$

where K_i is the pointwise stabilizer of $1, 2, \ldots, i-1$ in K. The algorithm is easiest to describe using the associated search tree, parts of which will, however, not be traversed because of pruning. The nodes of the tree are three-tuples of the form $(\mathcal{X}, K_i k, W)$, where $\mathcal{X} \subseteq \Omega$, $K_i k$ is a right coset of K_i in K, and W is an ordered partition of Π. A node is a leaf node if $i = n+1$.

The search tree $T(\mathcal{X}, K)$ is inductively defined as follows:

(5.3) $(\mathcal{X}, K_1, e(\mathcal{X}, (\Pi)))$ is a node in $T(\mathcal{X}, K)$, and

(5.4) if $(\mathcal{X}, K_i k, W)$ is a nonleaf node in $T(\mathcal{X}, K)$, and W_j is the first cell in W such that $K_i k(W_j) \cap K_i(i)$ has the minimum size subject to being nonempty, then $(\mathcal{X}, K_{i+1} lk, e(\mathcal{X}, W \wedge (\{p\}, \Pi - \{p\})))$ is a node in $T(\mathcal{X}, K)$ for all $p \in \Pi$ and all right cosets $K_{i+1} l$ in K_i satisfying $W(p) = j$ and $i = lk(p)$.

From (5.2), (5.3), and (5.4), we obtain for any $g \in K$ that $(\mathcal{X}, K_i k, W)$ is a node of $T(\mathcal{X}, K)$ if and only if $(g\mathcal{X}, K_i k g^{-1}, gW)$ is a node of $T(g\mathcal{X}, K)$. In particular, $T(\mathcal{X}, K) = T(\alpha\mathcal{X}, K)$ for all $\alpha \in \mathrm{Aut}_K(\mathcal{X})$, which allows us to find generators for $\mathrm{Aut}_K(\mathcal{X})$ as we traverse the search tree in depth-first order. Furthermore, symmetric subtrees in the search tree can be pruned using discovered automorphisms. We omit the detailed description of how the automorphisms are discovered and applied during the traversal of the search tree $T(\mathcal{X}, K)$; the techniques used here are analogous to those applied in [53].

The canonical labeling permutation $\kappa(\mathcal{X})$ is the first permutation encountered at a leaf node of $T(\mathcal{X}, K)$ that takes $\mathcal{X}$ to the K-canonical representative of $\mathcal{X}$. We determine the K-canonical representative according to a procedure analogous to the use of *indicator functions* (search tree node invariants) in [53]; see also [57, sect. 7]. The node invariant that we apply is a hash digest obtained as a side effect of executing the refinement procedure for $(\mathcal{X}, W)$. This hash digest is isomorphism-invariant in the sense that an identical digest is obtained for all inputs $(\mathcal{X}, W)$ and $(g\mathcal{X}, gW)$, where $g \in K$.

6. The STS(21)s with a nontrivial automorphism group. We first determine the prime-order automorphisms admitted by an STS(21) and then proceed to construct an associated set of seeds.

A prime-order subgroup H of the symmetric group S_{21} is determined up to conjugacy in S_{21} by the cycle type of its nontrivial elements. We use exponential notation for the cycle type. For example, the cycle type $1^3 2^9$ indicates that the corresponding permutation factors into three fixed points and nine disjoint 2-cycles.

Not all prime-order subgroups of S_{21} can occur as a group of automorphisms of an STS(21). The following observation is well known.

Lemma 6.1. *Let $\mathcal{X}$ be an* STS(v) *and let $\alpha \in \mathrm{Aut}(\mathcal{X})$. Then, the set of fixed points of α induces a subsystem of $\mathcal{X}$.*

Proof. Let x, y be fixed points of α. Let $\{x, y, z\} \in \mathcal{X}$ be the block that contains the pair $\{x, y\}$. Because α is an automorphism, also $\{x, y, \alpha(z)\} \in \mathcal{X}$. Thus, $\alpha(z) = z$ because otherwise the pair $\{x, y\}$ occurs in more than one block in $\mathcal{X}$. It follows that any block in $\mathcal{X}$ intersects the set of fixed points of α in exactly 0, 1, or 3 points. □

In particular, if α has f fixed points, then $f = 0$, or $f \equiv 1 \pmod 6$, or $f \equiv 3 \pmod 6$. The following observation is implied by the results in [24, 62, 69].

Lemma 6.2. *A permutation with cycle type $1^1 2^{10}$ cannot occur as an automorphism of an* STS(21).

Proof. To reach a contradiction, let $\mathcal{X}$ be an STS(21) admitting an automorphism of type $1^1 2^{10}$, let 0 be the fixed point, and let $\{\{-i, i\} : 1 \le i \le 10\}$ be the pairs of points fixed by such an automorphism. Thus, $\{x, y, z\} \in \mathcal{X}$ if and only if $\{-x, -y, -z\} \in \mathcal{X}$. In particular, $\{0, -i, i\} \in \mathcal{X}$ for all $1 \le i \le 10$. The remaining 60 blocks in $\mathcal{X}$ must cover the remaining 90 pairs of points with opposite signs. Thus, each of the 30 remaining orbits of blocks in $\mathcal{X}$ covers either 0 or 4 of the remaining pairs of points with opposite signs. This is a contradiction because 4 does not divide 90. □

This leaves us with nine possible prime-order cycle types:

$$3^7, \quad 7^3, \quad 1^1 5^4, \quad 1^3 2^9, \quad 1^3 3^6, \quad 1^7 2^7, \quad 1^7 7^2, \quad 1^9 2^6, \quad 1^9 3^4.$$

The next step is to select suitable transversal invariants for constructing the seeds. Table 1 contains the transversal invariant employed for each cycle type. The column "Seeds" contains the number of nonisomorphic seeds found for each cycle type and transversal invariant. The column "Occurrences" is used in the double-counting consistency check in section 7.2.

Table 1
Seeds for STS(21)*s with a nontrivial automorphism group.*

Cycle type	Transversal invariant	Seeds	Occurrences
3^7	$I_1^{\mathrm{fix}(0)}$	107	21
7^3	$I_1^{\mathrm{fix}(0)}$	188	21
$1^1 5^4$	$I_2^{\mathrm{fix}(1)}$	1265	20
$1^3 2^9$	$I_3^{\mathrm{fix}(3)}$	7034	1
$1^3 3^6$	$I_3^{\mathrm{fix}(3)}$	57	1
$1^7 2^7$	$I_3^{\mathrm{fix}(3)} \cap I_3^{\mathrm{blk}}$	277	7
$1^7 7^2$	$I_3^{\mathrm{fix}(3)} \cap I_3^{\mathrm{blk}}$	2	7
$1^9 2^6$	$I_3^{\mathrm{fix}(3)} \cap I_3^{\mathrm{blk}}$	306	12
$1^9 3^4$	$I_3^{\mathrm{fix}(3)} \cap I_3^{\mathrm{blk}}$	18	12

Table 2
The STS(21)*s with a nontrivial automorphism group.*

$\lvert\mathrm{Aut}(\mathcal{X})\rvert$	STS(21)s	Anti-Pasch
2	60588267	123
3	1732131	792
4	11467	0
5	1772	24
6	2379	4
7	66	8
8	222	0
9	109	3
12	85	0
14	14	0
16	12	0
18	33	1
21	10	3
24	19	0
27	3	0
36	5	0
42	7	0
48	2	0
54	1	0
72	5	0
108	1	0
126	2	0
144	1	0
294	1	0
504	1	0
882	1	0
1008	1	0
Total	62336617	958

Our algorithm implementation computes the seeds in about 10 minutes on a Linux PC with a 2-GHz CPU.

By extending the seeds in all possible ways and rejecting isomorphs, we obtain a complete classification of the STS(21)s with a nontrivial automorphism group in about 25 hours on a Linux PC with a 2-GHz CPU.

Theorem 6.3. *The number of pairwise nonisomorphic* STS(21)*s with a nontrivial automorphism group is* 62336617*.*

Table 2 partitions the STS(21)s into classes based on the automorphism group order. Listed in the column "Anti-Pasch" is the number of STS(21)s that do not contain a Pasch configuration; see (5.1).

Table 3 partitions the STS(21)s into classes based on the order of the automorphism group and the types of prime-order automorphisms they admit.

In addition to the prime-order automorphisms, permutations with the following cycle types occur as automorphisms of STS(21)s:

$$\begin{array}{lllllll} 21^1, & 7^1 14^1, & 3^1 18^1, & 3^3 6^2, & 3^1 9^2, & 3^1 6^1 12^1, & 3^1 6^3, \\ 1^1 2^1 3^2 6^2, & 1^1 2^4 4^3, & 1^3 6^3, & 1^3 3^2 6^2, & 1^3 2^3 6^2, & 1^3 2^3 4^3. & \end{array}$$

No other cycle types occur as automorphisms.

7. Consistency checking. Due to the relatively complex techniques employed in the classification algorithm, there is a real possibility that the implementation—which at present contains approximately 10,000 lines of C code (not including the

TABLE 3
Prime-order automorphisms in STS(21)*s.*

Class	3^7	7^3	$1^1 5^4$	$1^3 2^9$	$1^3 3^6$	$1^7 2^7$	$1^7 7^2$	$1^9 2^6$	$1^9 3^4$	#
1008	*	*		*	*	*		*		1
882	*	*			*	*	*			1
504	*	*			*			*		1
294		*			*	*	*			1
144	*			*	*	*		*		1
126a	*	*		*	*					1
b	*	*			*	*				1
108				*	*	*		*	*	1
72a	*			*	*	*		*		4
b	*			*	*			*		1
54	*			*	*				*	1
48				*	*	*		*		2
42a	*	*		*						1
b		*			*	*				2
c					*	*	*			4
36a	*				*			*		4
b				*	*	*		*		1
27	*				*					3
24a	*			*		*		*		1
b	*			*				*		5
c	*							*		1
d				*	*	*		*		9
e				*	*			*		2
f					*			*		1
21a	*	*								4
b		*			*					4
c					*		*			2
18a	*			*	*					8
b	*				*	*			*	3
c	*				*	*				5
d				*	*					2
e					*	*				14
f					*			*	*	1
16				*		*		*		12
14		*				*				14
12a	*			*				*		4
b	*							*		24
c				*	*	*		*		6
d				*		*		*	*	11
e					*			*		40
9a	*				*				*	9
b	*				*					68
c					*					32
8a				*		*		*		143
b				*				*		56
c								*		23
7		*								66
6a	*			*						723
b	*					*				5
c	*							*		13
d				*	*					264
e				*					*	11
f					*	*				1263
g					*			*		35
h						*			*	1
i								*	*	64
5			*							1772
4a				*		*		*		4069
b				*				*		1940
c								*		5458
3a	*									553918
b					*					1178118
c									*	95
2a				*						46191977
2b						*				11199633
2c								*		3196657
Total	554811	97	1772	46199257	1179916	11205208	8	3208591	197	62336617

source code for *nauty*)—contains errors. Therefore, it is necessary to incorporate consistency checking mechanisms into the implementation to catch errors that affect classification results obtained. For a further discussion on consistency checking in classification algorithms, we refer the reader to [43].

The present algorithm implementation contains two fundamental components that should be subjected to consistency checks: generation and isomorph rejection.

Generation—that is, seed generation and seed extension—is achieved using our own implementation of the exact cover algorithm in [37]. This implementation has been extensively tested and subjected to the consistency checks in [31, 32, 33]. Thus, we have a relatively high degree of confidence in the correct operation of the implementation.

The most complex and hence error-prone parts of the implementation are the routines that perform isomorph rejection; cf. [44].

We perform consistency checks in the seed generation phase using two independent hash-accumulators for every transversal size (transversal layer) and every number of extending orbits (point completion layer). The first accumulator records the structures subjected to an isomorph rejection test. Here a structure is recorded regardless of whether it is accepted or rejected in the test. The second accumulator is updated only after a structure has been accepted. In this case, every structure from which the accepted structure "is supposed to originate" is recorded into the second accumulator. When the seed generation algorithm terminates, we check that the corresponding accumulator values are identical. This gives us confidence that the isomorph rejection routines in the seed generation phase exhibit proper behavior. A more detailed description of the seed generation consistency checks is given in section 7.1.

To catch errors in the isomorph rejection phase for target systems, we employ a double-counting technique analogous to the ones used in [31, 32, 33]. Namely, we count in two different ways the total number of labelled target systems with one seed individualized. The first count relies on the classification obtained for target systems; the second count relies on the automorphism groups of seeds and on the number of target systems that extend the seeds. This also provides a further consistency check for the exact cover algorithm implementation. The double-counting technique is described in section 7.2.

7.1. Consistency checks during seed generation. We begin by describing the structure of a hash-accumulator used in the consistency checks. We assume that the structures to be recorded are encoded as finite binary strings (sequences of bytes). We write $\{0,1\}^*$ for the set of all finite binary strings. A hash-accumulator maintains an N-bit value $z \in \{0,1\}^N$, where our implementation uses $N = 32$. Given a structure $s \in \{0,1\}^*$ to be recorded, the value z is updated by $z \leftarrow a(z,s)$, where $a : \{0,1\}^N \times \{0,1\}^* \to \{0,1\}^N$ satisfies

$$a(a(z,s),t) = a(a(z,t),s) \quad \text{for all } z \in \{0,1\}^N,\ s,t \in \{0,1\}^*. \tag{7.1}$$

Property (7.1) implies that the order in which the structures are recorded does not affect the resulting value z.

We implement a using a hash function $h : \{0,1\}^* \to \{0,1\}^N$ and put $a(z,s) = z + h(s)$, where "+" denotes binary N-bit addition. For an extensive discussion of hash functions, see [36, sect. 6.4]. To obtain a hash-accumulator with good error-detecting capabilities, it should be unlikely that the selected hash function h satisfies $\sum_{s\in S} h(s) = \sum_{s'\in S'} h(s')$ for two sets of structures S, S' unless $S = S'$. Our somewhat ad hoc hash function relies on look-up tables of 256 random 32-bit words to scatter the hash values to $\{0,1\}^{32}$.

A more powerful alternative to hash-accumulators would be to keep a complete record of the structures encountered; see [44].

7.1.1. Point completion layer. Before the isomorph rejection for $(T,\mathcal{S},p)$, we proceed as follows. Let $\mathcal{O} \in H\backslash\Omega$, $\mathcal{O} \subseteq \mathcal{S}$ be the most recently added extending orbit. We compute $(\hat{T},\hat{\mathcal{S}},\hat{p})$ as in Example 11 and put

$$\begin{aligned} \hat{\mathcal{O}}_1 &\leftarrow \kappa(\mathcal{R})\mathcal{O}, \\ \hat{\mathcal{O}} &\leftarrow \min\{\alpha\hat{\mathcal{O}}_1 : \alpha \in \mathrm{Aut}_{N_G(H)}(\hat{T},\hat{\mathcal{S}},\hat{p})\}, \end{aligned}$$

where the minimum is taken relative to the lexicographic order induced by the lexicographic order on Ω. We then accumulate the first consistency check hash by the four-tuple $(\hat{T},\hat{\mathcal{S}},\hat{p},\hat{\mathcal{O}})$, provided that we have not done so earlier during the current invocation of the point completion layer. We keep track of this by storing the accumulated four-tuples in the hash table Z.

If the isomorph rejection tests accept $(T,\mathcal{S},p)$, then we accumulate the second consistency hash by the four-tuple $(\hat{T},\hat{\mathcal{S}},\hat{p},\hat{\mathcal{O}})$ for every extending orbit $\hat{\mathcal{O}} \in H\backslash\Omega$, $\hat{\mathcal{O}} \subseteq \hat{\mathcal{S}}$ that satisfies

$$\hat{\mathcal{O}} = \min\{\alpha\hat{\mathcal{O}} : \alpha \in \mathrm{Aut}_{N_G(H)}(\hat{T},\hat{\mathcal{S}},\hat{p})\}.$$

We employ an array of such pairs of hash-accumulators, one pair for each possible number of extending orbits in $(T,\mathcal{S},p)$. Initially—that is, when the transversal layer invokes the point completion layer—all accumulators are set to zero. By the structure of the point completion layer, the hash-accumulators in every pair should have equal value when the control returns back to the transversal layer. Note that because of the alternating invocations of the transversal layer and the point completion layer, we have to keep a separate array of accumulators for each invocation of the point completion layer made by the transversal layer.

7.1.2. Transversal layer. Before the isomorph rejection for $(T,\mathcal{S})$, we proceed as follows. Let $p \in T$ be the most recently completed point. We compute $(\hat{T},\hat{\mathcal{S}})$ as in Example 20 and put

$$\begin{aligned} \hat{p}_1 &\leftarrow \kappa(\mathcal{R})p, \\ \hat{p} &\leftarrow \min\{\alpha\hat{p}_1 : \alpha \in \mathrm{Aut}_{N_G(H)}(\hat{T},\hat{\mathcal{S}})\}. \end{aligned}$$

We then accumulate the first consistency check hash by the three-tuple $(\hat{T},\hat{\mathcal{S}},\hat{p})$, provided that we have not done so earlier during the current invocation of the transversal layer. We keep track of this by storing the accumulated three-tuples in the hash table Y.

If the isomorph rejection tests accept $(T,\mathcal{S})$, then we accumulate the second consistency hash by the three-tuple $(\hat{T},\hat{\mathcal{S}},\hat{p})$ for every parent point $\hat{p} \in \mathrm{par}_{|\hat{T}|}(\hat{T},\hat{\mathcal{S}})$ that satisfies

$$\hat{p} = \min\{\alpha\hat{p} : \alpha \in \mathrm{Aut}_{N_G(H)}(\hat{T},\hat{\mathcal{S}})\}.$$

Again we employ an array of hash-accumulators, one pair for each possible transversal size. All accumulators are set to zero during the initialization of the seed generation algorithm for a given $H \in \mathcal{H}$. By the structure of the transversal layer, the values in every accumulator pair should be equal when the seed generation for H is complete.

7.2. Double-counting consistency check. To check consistency of the isomorph rejection for target systems, we employ the following double-counting technique for every $H \in \mathcal{H}$.

First, for every H-seed $(H, T, \mathcal{S})$, we compute the number $\#\mathrm{ext}(H, T, \mathcal{S})$ of target systems that extend $(H, T, \mathcal{S})$. This number is easily obtained for every H-seed as a side effect of the seed extension phase.

Second, for every target system $\mathcal{X}$ that is accepted in the isomorph rejection tests, we compute the number of H-seeds occurring in $\mathcal{X}$. More precisely, let $I_0, I_1, \ldots, I_k$ be the sequence of transversal invariants used in computing the H-seeds. Extend I_k to conjugates of H by (4.6). Let $\#\mathrm{seeds}(H, \mathcal{X})$ be the number of three-tuples $(H_0, T_0, \mathcal{S}_0)$, where $H_0 \leq \mathrm{Aut}(\mathcal{X})$, H_0 is conjugate to H, $T_0 \in I_k(H_0, \mathcal{S}_0)$, and $\mathcal{S}_0 = (H_0, T_0)\downarrow\mathcal{X}$. Each such three-tuple is an H-seed occurring in $\mathcal{X}$.

For prime-order cyclic groups H, we compute $\#\mathrm{seeds}(H, \mathcal{X})$ by simple exhaustive search on $\mathrm{Aut}(\mathcal{X})$. Whenever we encounter a $g \in \mathrm{Aut}(\mathcal{X})$ that is in the same conjugacy class as the nontrivial elements of H, we count the number of admissible sets T_0 for the group $H_0 = \langle\{g\}\rangle$. Often this number is a constant that can be determined by combinatorial arguments based on property P and the group H; the column "Occurrences" in Table 1 lists the number of admissible sets T_0 for the seeds used in the STS(21) classification. After the exhaustive search terminates, we divide the total count by $|H| - 1$ and obtain $\#\mathrm{seeds}(H, \mathcal{X})$.

We now obtain a double-counting argument as follows. Let $\mathscr{S}(H)$ be the set of all H-seeds output by the seed generation algorithm, and let $\mathscr{X}$ be the set of all target systems output by the top-level algorithm. By the orbit-stabilizer theorem, we should obtain for every $H \in \mathcal{H}$

$$\sum_{(H,T,\mathcal{S})\in\mathscr{S}(H)} \frac{|G| \cdot \#\mathrm{ext}(H, T, \mathcal{S})}{|\mathrm{Aut}(H, T, \mathcal{S})|} = \sum_{\mathcal{X}\in\mathscr{X}} \frac{|G| \cdot \#\mathrm{seeds}(H, \mathcal{X})}{|\mathrm{Aut}(\mathcal{X})|}, \tag{7.2}$$

where both sides of the equation enumerate the total number of target systems with one H-seed individualized.

For the STS(21)s admitting a nontrivial group of automorphisms, both sides of (7.2) evaluate to identical values for every eligible group H. This—together with the fact that the consistency checks in the seed generation algorithm detect no errors—gives us confidence that the classification reported in section 6 is correct. The values of (7.2) obtained for STS(21)s appear in Table 4.

Table 4
Consistency check counts for STS(21)*s.*

Cycle type	Consistency check count
3^7	198275056744466459197440000
7^3	11741185091460464640000
$1^1 5^4$	362132598113076510720000
$1^3 2^9$	1180117747062230388572160000
$1^3 3^6$	20079376071873280081920000
$1^7 2^7$	2003157311566888776499200000
$1^7 7^2$	72987060245299200000
$1^9 2^6$	983300360312093618995200000
$1^9 3^4$	28610927616157286400000

8. Discussion. We have also tested the algorithm implementation on two other families of designs that are closely related to Steiner triple systems, namely, one-factorizations of complete graphs and Latin squares, both of which can be represented as certain triple systems. With minor modifications, the present algorithm implementation produces a correct classification of the one-factorizations of the complete graph K_{12} [22] (1.5 hours on a 2-GHz Linux PC) and Latin squares of side 9 [56] (16 hours on a 2-GHz Linux PC), where both classifications are restricted to systems admitting a nontrivial automorphism group.

In the case of one-factorizations of K_{14}, a crude estimate indicates that a complete classification of the systems admitting a nontrivial automorphism group would require about half a year on a 2-GHz Linux PC. Performing this classification is a topic of future work.

Despite these successful applications, however, it should be made clear that the applicability of the present classification approach is restricted by a number of factors.

Perhaps the most fundamental restricting factor is that the approach requires the existence of a suitable collection of seeds. For triple systems, the present type of seed appears to perform well, but the performance for larger block sizes and other families of designs remains to be studied.

Another restricting factor is that explicit isomorphism computations are required, which makes the approach applicable only to parameter values small enough to admit practical isomorphism computations for seeds and target systems. For large parameter values it appears that implicit group-theoretic techniques [46] must be used.

Furthermore, it should be observed that here we have considered only prime-order prescribed groups. More complex groups can be used, but this requires the implementation of more complex distinguishing primitives during the final isomorph rejection phase, which presents a possible performance bottleneck.

Acknowledgments. The author would like to thank Harri Haanpää, Clement Lam, and Patric Östergård for useful discussions.

REFERENCES

[1] L. Babai, *Automorphism groups, isomorphism, reconstruction*, in Handbook of Combinatorics, Vol. 2, R. L. Graham, M. Grötschel, and L. Lovász, eds., North-Holland, Amsterdam, 1995, pp. 1447–1540.

[2] T. Beth, D. Jungnickel, and H. Lenz, *Design Theory*, Vols. 1 and 2, 2nd ed., Cambridge University Press, Cambridge, UK, 1999.

[3] A. Betten, A. Kerber, A. Kohnert, R. Laue, and A. Wassermann, *The discovery of simple* 7*-designs with automorphism group* $\mathrm{P\Gamma L}(2,32)$, in Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, G. Cohen, M. Giusti, and T. Mora, eds., Lecture Notes in Comput. Sci. 948, Springer, Berlin, 1995, pp. 131–145.

[4] A. Betten, R. Laue, S. Molodtsov, and A. Wassermann, *Steiner systems with automorphism groups* $\mathrm{PSL}(2,71)$*,* $\mathrm{PSL}(2,83)$*, and* $\mathrm{P\Sigma L}(2,3^5)$, J. Geom., 67 (2000), pp. 35–41.

[5] A. Betten, R. Laue, and A. Wassermann, *Simple* 7*-designs with small parameters*, J. Combin. Des., 7 (1999), pp. 79–94.

[6] G. Brinkmann, *Isomorphism rejection in structure generation programs*, in Discrete Mathematical Chemistry, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 51, P. Hansen, P. Fowler, and M. Zheng, eds., AMS, Providence, RI, 2000, pp. 25–38.

[7] G. Brinkmann and B. D. McKay, *Posets on up to* 16 *points*, Order, 19 (2002), pp. 147–179.

[8] C. A. Brown, L. Finkelstein, and P. W. Purdom, Jr., *A new base change algorithm for permutation groups*, SIAM J. Comput., 18 (1989), pp. 1037–1047.

[9] G. Butler, *Fundamental Algorithms for Permutation Groups*, Lecture Notes in Comput. Sci. 559, Springer, Berlin, 1991.

[10] G. Butler and C. W. H. Lam, *A general backtrack algorithm for the isomorphism problem of combinatorial objects*, J. Symbolic Comput., 1 (1985), pp. 363–381.

[11] M. B. Cohen, C. J. Colbourn, L. A. Ives, and A. C. H. Ling, *Kirkman triple systems of order* 21 *with a nontrivial automorphism group*, Math. Comp., 71 (2002), pp. 873–881.

[12] C. J. Colbourn and J. H. Dinitz, eds., *The CRC Handbook of Combinatorial Designs*, CRC Press Ser. Discrete Math. Appl., CRC Press, Boca Raton, FL, 1996.

[13] C. J. Colbourn, S. S. Magliveras, and D. R. Stinson, *Steiner triple systems of order* 19 *with nontrivial automorphism group*, Math. Comp., 59 (1992), pp. 283–295.

[14] C. J. Colbourn and A. Rosa, *Triple Systems*, Oxford Math. Monogr., Clarendon Press, Oxford, UK, 1999.

[15] M. J. Colbourn and R. A. Mathon, *On cyclic Steiner* 2*-designs*, Ann. Discrete Math., 7 (1980), pp. 215–253.

[16] G. Cooperman and L. Finkelstein, *A fast cyclic base change for permutation groups*, in Papers from the International Symposium on Symbolic and Algebraic Computation (ISSAC'92, Berkeley, CA, 1992), ACM Press, New York, 1992, pp. 224–232.

[17] D. Crnković, *Symmetric* $(70, 24, 8)$ *designs having* $\mathrm{Frob}_{21} \times Z_2$ *as an automorphism group*, Glas. Mat. Ser. III, 34(54) (1999), pp. 109–121.

[18] P. Dembowski, *Verallgemeinerungen von Transitivitätsklassen endlicher projektiver Ebenen*, Math. Z., 69 (1958), pp. 59–89.

[19] P. Dembowski, *Finite Geometries*, Springer, Berlin, 1997.

[20] M. Deza, *Une propriété extrémale des plans projectifs finis dans une classe de codes équidistants*, Discrete Math., 6 (1973), pp. 343–352.

[21] M. Deza, *Solution d'un probléme de Erdős–Lovász*, J. Combin. Theory Ser. B, 16 (1974), pp. 166–167.

[22] J. H. Dinitz, D. K. Garnick, and B. D. McKay, *There are* 526,915,620 *nonisomorphic one-factorizations of* K_{12}, J. Combin. Des., 2 (1994), pp. 273–285.

[23] J. D. Dixon and B. Mortimer, *Permutation Groups*, Springer, New York, 1996.

[24] J. Doyen, *A note on reverse Steiner triple systems*, Discrete Math., 1 (1972), pp. 315–319.

[25] Z. Eslami and G. B. Khosrovshahi, *A complete classification of* 3-$(11, 4, 4)$ *designs with nontrivial automorphism group*, J. Combin. Des., 8 (2000), pp. 419–425.

[26] I. A. Faradžev, *Constructive enumeration of combinatorial objects*, in Problèmes Combinatoires et Théorie des Graphes, Colloq. Internat. CNRS 260, CNRS, Paris, 1978, pp. 131–135.

[27] The GAP Group, GAP—*Groups, Algorithms, and Programming*, Version 4.4, http://www.gap-system.org (2004).

[28] P. B. Gibbons, *Computational methods in design theory*, in The CRC Handbook of Combinatorial Designs, CRC Press Ser. Discrete Math. Appl., C. J. Colbourn and J. H. Dinitz, eds., CRC Press, Boca Raton, FL, 1996, pp. 718–740.

[29] E. Haberberger, A. Betten, and R. Laue, *Isomorphism classification of t-designs with group theoretical localisation techniques applied to some Steiner quadruple systems on* 20 *points*, Congr. Numer., 142 (2000), pp. 75–96.

[30] D. Held and M.-O. Pavčević, *Symmetric* $(79, 27, 9)$*-designs admitting a faithful action of a Frobenius group of order* 39, European J. Combin., 18 (1997), pp. 409–416.

[31] P. Kaski and P. R. J. Östergård, *The Steiner triple systems of order* 19, Math. Comp., 73 (2004), pp. 2075–2092.

[32] P. Kaski and P. R. J. Östergård, *One-factorizations of regular graphs of order* 12, Electron. J. Combin., 12 (2005), Research Paper 2.

[33] P. Kaski, P. R. J. Östergård, S. Topalova, and R. Zlatarski, *Steiner triple systems of order* 19 *and* 21 *with subsystems of order* 7, Discrete Math., to appear.

[34] A. Kerber, *Applied Finite Group Actions*, 2nd ed., Algorithms Combin. 19, Springer, Berlin, 1999.

[35] G. B. Khosrovshahi, M. Mohammad-Noori, and B. Tayfeh-Rezaie, *Classification of* 6-$(14, 7, 4)$ *designs with nontrivial automorphism groups*, J. Combin. Des., 10 (2002), pp. 180–194.

[36] D. E. Knuth, *The Art of Computer Programming. Vol.* 3*. Sorting and Searching*, 2nd ed., Addison-Wesley, Reading, MA, 1998.

[37] D. E. Knuth, *Dancing links*, in Millennial Perspectives in Computer Science, J. Davies, B. Roscoe, and J. Woodcock, eds., Palgrave Macmillan, Basingstoke, UK, 2000, pp. 187–214.

[38] E. S. Kramer, S. S. Magliveras, and R. Mathon, *The Steiner systems* $S(2, 4, 25)$ *with nontrivial automorphism group*, Discrete Math., 77 (1989), pp. 137–157.

[39] E. S. Kramer and D. M. Mesner, *t-designs on hypergraphs*, Discrete Math., 15 (1976), pp. 263–296.

[40] D. L. Kreher and S. P. Radziszowski, *Finding simple t-designs by using basis reduction*, Congr. Numer., 55 (1986), pp. 235–244.

[41] D. L. Kreher and S. P. Radziszowski, *The existence of simple* 6-(14, 7, 4) *designs*, J. Combin. Theory Ser. A, 43 (1986), pp. 237–243.

[42] D. L. Kreher and S. P. Radziszowski, *Simple* 5-$(28, 6, \lambda)$ *designs from* $\mathrm{PSL}_2(27)$, Ann. Discrete Math., 37 (1987), pp. 315–318.

[43] C. W. H. Lam, *How reliable is a computer-based proof?* Math. Intelligencer, 12 (1990), pp. 8–12.

[44] C. W. H. Lam and L. Thiel, *Backtrack search with isomorph rejection and consistency check*, J. Symbolic Comput., 7 (1989), pp. 473–485.

[45] R. Laue, *Constructing objects up to isomorphism, simple* 9*-designs with small parameters*, in Algebraic Combinatorics and Applications, A. Betten, A. Kohnert, R. Laue, and A. Wassermann, eds., Springer, Berlin, 2001, pp. 232–260.

[46] R. Laue, *Solving isomorphism problems for t-designs*, in Designs, 2002, Math. Appl. 563, Kluwer Academic Publishers, Boston, MA, 2003, pp. 277–300.

[47] J. S. Leon, *Permutation group algorithms based on partitions.* I. *Theory and algorithms*, J. Symbolic Comput., 12 (1991), pp. 533–583.

[48] J. S. Leon, *Partitions, refinements, and permutation group computation*, in Groups and Computation, II, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 28, L. Finkelstein and W. M. Kantor, eds., AMS, Providence, RI, 1997, pp. 123–158.

[49] M. M-Noori and B. Tayfeh-Rezaie, *Backtracking algorithm for finding t-designs*, J. Combin. Des., 11 (2003), pp. 240–248.

[50] R. Mathon, *Symmetric* (31, 10, 3) *designs with nontrivial automorphism group*, Ars Combin., 25 (1988), pp. 171–183.

[51] R. Mathon, *Computational methods in design theory*, in Surveys in Combinatorics, 1991, London Math. Soc. Lecture Note Ser. 166, A. D. Keedwell, ed., Cambridge University Press, Cambridge, UK, 1991, pp. 101–117.

[52] R. A. Mathon, K. T. Phelps, and A. Rosa, *A class of Steiner triple systems of order* 21 *and associated Kirkman systems*, Math. Comp., 37 (1981), pp. 209–222; and 64 (1995), pp. 1355–1356 (addendum).

[53] B. D. McKay, *Practical graph isomorphism*, Congr. Numer., 30 (1981), pp. 45–87.

[54] B. D. McKay, *Nauty User's Guide* (*Version* 1.5), Technical report TR-CS-90-02, Computer Science Department, Australian National University, Canberra, Australia, 1990.

[55] B. D. McKay, *Isomorph-free exhaustive generation*, J. Algorithms, 26 (1998), pp. 306–324.

[56] B. D. McKay, A. Meynert, and W. Myrvold, *Small Latin Squares, Quasigroups and Loops*, preprint.

[57] T. Miyazaki, *The complexity of McKay's canonical labeling algorithm*, in Groups and Computation, II, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 28, L. Finkelstein and W. M. Kantor, eds., AMS, Providence, RI, 1997, pp. 239–256.

[58] P. R. J. Östergård, *Constructing combinatorial objects via cliques*, in Surveys in Combinatorics, 2005, London Math. Soc. Lecture Note Ser. 327, B. S. Webb, ed., Cambridge University Press, Cambridge, UK, 2005, pp. 57–82.

[59] M.-O. Pavčević, *Symmetric designs of Menon series admitting an action of Frobenius groups*, Glas. Mat. Ser. III, 31(51) (1996), pp. 209–223.

[60] S. Pfaff, *Classification of* (78, 22, 6) *designs having the full automorphism group* $E_8 \cdot F_{21}$, Glas. Mat. Ser. III, 28(48) (1993), pp. 3–9.

[61] R. C. Read, *Every one a winner; or, how to avoid isomorphism search when cataloguing combinatorial configurations*, Ann. Discrete Math., 2 (1978), pp. 107–120.

[62] A. Rosa, *On reverse Steiner triple systems*, Discrete Math., 2 (1972), pp. 61–71.

[63] J. J. Rotman, *An Introduction to the Theory of Groups*, 4th ed., Grad. Texts in Math. 148, Springer, New York, 1995.

[64] B. Schmalz, *The t-designs with prescribed automorphism group, new simple* 6*-designs*, J. Combin. Des., 1 (1993), pp. 125–170.

[65] E. Seah and D. R. Stinson, *On the enumeration of one-factorizations of complete graphs containing prescribed automorphism groups*, Math. Comp., 50 (1988), pp. 607–618.

[66] Á. Seress, *Permutation Group Algorithms*, Cambridge Tracts in Math. 15, Cambridge University Press, Cambridge, UK, 2003.

[67] E. Spence, *Symmetric* (31, 10, 3)*-designs with a nontrivial automorphism of odd order*, J. Combin. Math. Combin. Comput., 10 (1991), pp. 51–64.

[68] D. R. Stinson, *A comparison of two invariants for Steiner triple systems*: *Fragments and trains*, Ars Combin., 16 (1983), pp. 69–76.

[69] L. Teirlinck, *The existence of reverse Steiner triple systems*, Discrete Math., 6 (1973), pp. 301–302.

[70] V. D. Tonchev, *Steiner triple systems of order* 21 *with automorphisms of order* 7, Ars Combin., 23 (1987), pp. 93–96; and 39 (1995), p. 3 (erratum).

[71] S. Topalova, *Symmetric* 2-(69, 17, 4) *designs with automorphisms of order* 13, J. Statist. Plann. Inference, 95 (2001), pp. 335–339.

[72] S. Topalova, *Classification of Hadamard matrices of order* 44 *with automorphisms of order* 7, Discrete Math., 260 (2003), pp. 275–283.

[73] A. Wassermann, *Finding simple t-designs with enumeration techniques*, J. Combin. Des., 6 (1998), pp. 79–90.

SIAM J. Discrete Math.
Vol. 19, No. 3, pp. 691–699

THE ASYMPTOTIC NUMBER OF BINARY CODES AND BINARY MATROIDS[*]

MARCEL WILD[†]

Abstract. The asymptotic number of nonequivalent binary n-codes is determined. This is also the asymptotic number of nonisomorphic binary matroids on n elements.

Key words. asymptotic enumeration, binary codes, binary matroids, lattice of invariant subspaces

AMS subject classifications. Primary, 94A10, 05A16; Secondary, 05B35, 05A30

DOI. 10.1137/S0895480104445538

1. Introduction.

Recall that a *binary n-code* is a subspace X of the $GF(2)$-vector space $V := GF(2)^n$. Two binary n-codes $X, X' \subseteq V$ are *equivalent* if for some permutation σ of the symmetric group S_n on $\{1, 2, \ldots, n\}$ we have

$$X' = X_\sigma := \{(x_{1\sigma}, \ldots, x_{n\sigma}) |\ (x_1, \ldots, x_n) \in X\},$$

where $i\sigma$ is the image of i under σ. Let $b(n)$ be the number of equivalence classes of binary n-codes. It is well known that $b(n)$ is also the number of nonisomorphic binary matroids on an n-set. The asymptotic behavior of $b(n)$ was posed as open problem 14.5.4 in [O].

Here the setting of binary codes suits us better. For a field K let $G(n, K)$ be the (possibly infinite) number of K-linear subspaces of K^n. Mostly, K will be $GF(q)$, in which case we write $G(n, q)$ instead of $G(n, K)$. Because each equivalence class of binary n-codes has cardinality at most $n!$ it follows that $b(n) \geq G(n, 2)/n!$ for all n. It will be a corollary of our main theorem that for $n \to \infty$ asymptotically

$$b(n) \sim G(n, 2)/n!. \tag{1}$$

For $\sigma \in S_n$ let $T_\sigma : V \to V$ be the vector space automorphism defined on the canonical base by $T_\sigma(e_i) := e_{i\sigma}$. Let $\mathcal{L}(T_\sigma)$ be the lattice of all T_σ-*invariant* subspaces in the sense of linear algebra, meaning the lattice of all subspaces U with $T_\sigma(U) \subseteq U$. Since here T_σ is bijective, $T_\sigma(U) \subseteq U$ is equivalent to $T_\sigma(U) = U$, i.e., to U being a "fixed point." This allows us to apply the Cauchy–Frobenius lemma (erroneously called Burnside's lemma):

$$b(n) \quad = \quad \frac{G(n, 2)}{n!} + \frac{1}{n!} \sum_{\sigma \in S_n - \{id\}} |\mathcal{L}(T_\sigma)|, \tag{2}$$

hence proving (1) is equivalent to showing

$$\sum_{\sigma \in S_n - \{id\}} |\mathcal{L}(T_\sigma)| \quad = \quad o(G(n, 2)). \tag{3}$$

[*] Received by the editors August 6, 2004; accepted for publication (in revised form) April 4, 2005; published electronically November 23, 2005.
http://www.siam.org/journals/sidma/19-3/44553.html

[†] Department of Mathematics, University of Stellenbosch, 7602 Matieland, South Africa (mwild@sun.ac.za).

There are $\binom{n}{2}$ permutations $\tau \in S_n$ with one 2-cycle and $n-2$ cycles of length 1. Any such transposition τ yields a T_τ with at least $G(n-1,2)$ invariant subspaces. Indeed, say T_τ switches e_1 and e_2. Then the $n-1$ vectors $e_1+e_2,\ e_3,\dots,e_n$ are fixed by T_τ. Hence

$$\binom{n}{2}G(n-1,2) \quad \text{is a } \textit{lower} \text{ bound for} \sum_{\sigma\in S_n-\{id\}} |\mathcal{L}(T_\sigma)|. \tag{4}$$

This shows that (3) can only be true if $G(n,2)$ grows superexponentially with n. Proving (3) was undertaken in [W1] but, as pointed out by Lax [L], there is an error in the proof of [W1, Lemma 6]. The error is fixed in the present article, which also improves upon style and organization. In fact, we shall wind up with a stronger result but as a golden thread it may be helpful, at least in section 2, to think of (3) as our target. The stronger result consists of rather sharp lower and upper bounds for $b(n)$ when n is large enough. These bounds are derived in sections 3 and 4, respectively, and the pieces are put together in section 5.

2. Four lemmata. The first lemma reduces our preliminary target (3) to the statement that the left-hand side of (3) is $o(2^{n^2/4})$.

LEMMA 1. *For all prime powers q there are positive constants $d_1(q)$ and $d_2(q)$ such that*

$$\lim_{m\to\infty}\frac{G(2m+1,q)}{q^{(2m+1)^2/4}} = d_1(q) \quad \text{and} \quad \lim_{m\to\infty}\frac{G(2m,q)}{q^{(2m)^2/4}} = d_2(q).$$

Furthermore, all $d_i(q)$ are less than 32 *and rounded to six decimals, $d_1(2)$ is* 7.371949, *and $d_2(2)$ is* 7.371969.

Proof. Let q be fixed and put $G_n := G(n,q)$. Note that $G_0 = 1,\ G_1 = 2$. By [A, p. 94] one has

$$G_{n+1} \;=\; 2G_n + (q^n-1)G_{n-1} \quad (n\ge 1). \tag{5}$$

Letting $u_n := q^{-n^2/4}G_n$ $(n \ge 0)$, it follows from (5) that

$$u_n \;=\; 2q^{-n/2+1/4}u_{n-1} + (1-q^{-n+1})u_{n-2} \quad (n\ge 2). \tag{6}$$

Letting $\tau_n = \tau_n(q) := 2q^{-n/2+1/4} + 1 - q^{-n+1},\ a_n := 2q^{-n/2+1/4}\tau_n^{-1}$, and $b_n := (1-q^{-n+1})\tau_n^{-1}$, we have $a_n + b_n = 1$ and

$$u_n \;=\; \tau_n(a_nu_{n-1} + b_nu_{n-2}) \quad (n\ge 2). \tag{7}$$

From $u_0 = 1,\ u_1 = 2q^{-1/4},\ \tau_n > 1$ $(n \ge 2)$, and (7) it follows that

$$u_n \;\ge\; \min\{u_0,u_1\} \;>\; 0 \quad (n\ge 0). \tag{8}$$

As to an upper bound, from $u_0 = 1$ and $u_1 \le 2\cdot 2^{-1/4} < 1.7$ we get $a_2u_1 + b_2u_0 \le 1.7$, so (7) yields $u_2 \le (1.7)\tau_2,\ u_3 \le (1.7)\tau_2\tau_3$, and so forth. One checks that $\tau_n(q) \le \tau_n(2)$ for $n \ge 2$ and $\tau_n(2) \le 1 + 2^{-n/3}$ for $n \ge 7$, whence

$$u_n \;\le\; (1.7)\tau_2(2)\cdots\tau_6(2)\cdot\prod_{k\ge 7}(1+2^{-k/3}) \;<\; (1.7)\cdot(6.8)\cdot e^{0.97} \;<32 \quad (n\ge 0). \tag{9}$$

The convergence of the latter infinite product follows by taking natural logarithms and noticing that $\sum_{k\geq 7} ln(1+2^{-k/3})$ is bounded by $\sum_{k\geq 7} 2^{-k/3} \; < \; 0.97$. From (6) and (9) it follows that

$$|u_{n+2}-u_n| \;=\; |-q^{-n-1}u_n + 2q^{-\frac{n}{2}-\frac{3}{4}}u_{n+1}| \;\leq\; 32q^{-\frac{n}{3}} \quad (n\geq 0).$$

Iterating and applying the triangle inequality yields

$$|u_{n+2k}-u_n| \;\leq\; 32(q^{-\frac{n}{3}}+q^{-\frac{(n+2)}{3}}+\cdots+q^{-\frac{(n+2k-2)}{3}}) \;\leq\; 32\frac{q^{-\frac{n}{3}}}{1-q^{-\frac{2}{3}}} \quad (n\geq 0). \tag{10}$$

Cauchy's criterion therefore guarantees that both $d_1(q) := \lim_{m\to\infty} u_{2m+1}$ and $d_2(q) := \lim_{m\to\infty} u_{2m}$ exist. They are nonzero by (8). Clearly, (10) implies

$$|d_2(q)-u_{2m}| \;\leq\; 32\;\frac{q^{-\frac{2m}{3}}}{1-q^{-\frac{2}{3}}} \quad (m\geq 0). \tag{11}$$

Combining (7) and (11), one can compute $d_2(q)$ to any desired accuracy. Ditto for $d_1(q)$. □

In order to get a handle on $\mathcal{L}(T_\sigma)$ we need the minimal polynomial

$$\min(T_\sigma,t) = \prod_{i=1}^{s} p_i(t)^{\mu_i},$$

where the $p_i(t) \in GF(2)[t]$ are irreducible and $\mu_i \geq 1$ $(1\leq i\leq s)$. We seek an upper bound for $s = s(\sigma)$. Since $\min(T_\sigma,t)$ has degree at most n and since there are only finitely many irreducible polynomials in $GF(2)[t]$ of any given degree, it is clear that for any fixed $\epsilon > 0$ one can force $s \leq \epsilon n$ for all $\sigma \in S_n$, provided n is large enough. For us it will suffice that

(12) for all large enough n one has $s \leq (0.06)n$ for all $\sigma \in S_n$.

It is well known that if

$$V_i := \ker(p_i(T_\sigma)^{\mu_i}), \quad n_i := \dim(V_i) \quad (1\leq i\leq s),$$

then $V = V_1 \oplus \cdots \oplus V_s$; and if $T_i := (T_\sigma \restriction V_i)$, then $T_i : V_i \to V_i$ has minimal polynomial $\min(T_i,t) = p_i(t)^{\mu_i}$. Furthermore, by [BF, p. 812]

$$\mathcal{L}(T_\sigma) \quad\simeq\quad \mathcal{L}(T_1)\times\mathcal{L}(T_2) \quad\times\cdots\times\quad \mathcal{L}(T_s). \tag{13}$$

Assume that our σ is a product of r disjoint cycles $C_1,\dots,C_r$ of lengths $\lambda_j = 2^{\alpha_j}\cdot u_j$, where $\alpha_j \geq 0$ and $u_j \geq 1$ is odd. The upcoming (14) and (15) will be the only facts for which we refer to [W1]. Namely, if we standardize $p_1(t) := t+1$, then its corresponding parameters μ_1 and n_1 satisfy [W1, Lemma 4]

$$\mu_1 = \max\{2^{\alpha_j} \mid 1\leq j\leq r\} \tag{14}$$

and [W1, Lemma 5]

$$r \quad\leq\quad n_1 = 2^{\alpha_1}+2^{\alpha_2}+\cdots+2^{\alpha_r} \quad\leq\quad n. \tag{15}$$

For instance, $\sigma := (1,2,\ldots,11,12)(13,14,15)(16,17)$ has $n_1 = 2^2+2^0+2^1 = 7$ and a base of V_1 is

$$e_1+e_5+e_9,\ e_2+e_6+e_{10},\ e_3+e_7+e_{11},\ e_4+e_8+e_{12},\ e_{13}+e_{14}+e_{15},\ e_{16},\ e_{17}.$$

Observe that while $\min(T_\sigma,t)$ is just the least common multiple of the polynomials $t^{\lambda_j}+1$ $(1 \le j \le r)$, the prime factors of $\min(T_\sigma,t)$ are unpredictable, and hence there is no general connection between the number r of disjoint cycles of σ and the number s of direct factors of $\mathcal{L}(T_\sigma)$. It is well known that the expected value of $r(\sigma)$ asymptotically is $ln(n)$ as $n \to \infty$. Question: What is the expected value of $s(\sigma)$ as $n \to \infty$?

LEMMA 2. *For large enough n all $\sigma \in S_n$ have $|\mathcal{L}(T_\sigma)| \le |\mathcal{L}(T_1)| \cdot 2^{\frac{(n-n_1)^2}{8}+(0.3)n}$.*

Proof. Since T_i is bijective we have $T_i^{\mu_i} \ne 0$, so $p_i(t) = t$ is impossible, so each $p_i(t)$ $(2 \le i \le s)$ has degree $d_i \ge 2$. Fix $T_i : V_i \to V_i$ with $2 \le i \le s$. According to [BF, Thm. 6] one can write $T_i = Q+S$, where $S : V_i \to V_i$ is semisimple and $Q : V_i \to V_i$ is nilpotent. Moreover, putting $K := GF(2)[t]/p_i(t)$, the map Q is K-linear in a natural sense and $\mathcal{L}(T_i) = \mathcal{L}_K(Q)$. Since $K \simeq GF(2^{d_i})$ and $\dim_K(V_i) = n_i/d_i$, it follows from Lemma 1 that

$$|\mathcal{L}(T_i)| \quad \le \quad G\left(\frac{n_i}{d_i}, 2^{d_i}\right) \quad \le \quad 2^5 \cdot (2^{d_i})^{(n_i/d_i)^2/4} \quad = \quad 2^5 \cdot 2^{n_i^2/4d_i}.$$

Using (12) and $d_i \ge 2$ $(2 \le i \le s)$ we get

$$\begin{aligned} |\mathcal{L}(T_\sigma)| &\le |\mathcal{L}(T_1)|(2^5 \cdot 2^{n_2^2/8}) \cdots (2^5 \cdot 2^{n_s^2/8}) \\ &\le |\mathcal{L}(T_1)| \cdot (2^5)^{(0.06)n} \cdot 2^{n_2^2/8+\cdots+n_s^2/8} \\ &\le |\mathcal{L}(T_1)| \cdot 2^{(0.3)n+(n_2+\cdots+n_s)^2/8}. \qquad \square \end{aligned}$$

The trick to decompose T_i as $S+Q$ with Q nilpotent and $\mathcal{L}(T_i) = \mathcal{L}_K(Q)$ also works for $T_i = T_1$. In fact one verifies at once that $T_1 = I+(T_1+I)$ with $(T_1+I)^{\mu_1} = 0$ and $\mathcal{L}(T_1) = \mathcal{L}(T_1+I)$. However, $d_i \ge 2$ is essential in the proof of Lemma 2; for $d_1 = 1$ one only gets the triviality (in view of Lemma 1) $|\mathcal{L}(T_1)| = O(2^{n_1^2/4})$. On the other hand, information about $\ker(Q)$ is only available when $i = 1$, and that is what makes the next lemma tick.

LEMMA 3. *Let $\sigma \in S_n$ have r disjoint cycles. With T_1, n_1, μ_1 derived from T_σ as above, one has*

(a) $|\mathcal{L}(T_1)| \quad \le \quad G(r,2) \cdot G(n_1-r,2)$,

(b) $|\mathcal{L}(T_1)| \quad \le \quad G(r,2)^{\mu_1}$.

Proof. Let W be any K-vector space with $\dim(W) = \overline{n}$ and $Q : W \to W$ a linear nilpotent map, say $Q^{m-1} \ne Q^m = 0$. Let $Q_2 := Q \restriction \text{im }(Q)$. Note that $Q_2 \ne Q^2$ but $\text{im }(Q_2) = \text{ im }(Q^2)$. It is easy to see [BF, Thm. 7] that

$$\mathcal{L}(Q) = \bigcup_{X \in \mathcal{L}(Q_2)} [X, Q^{-1}(X)], \tag{16}$$

where $Q^{-1}(X) := \{w \in W |\ Q(w) \in X\}$ and $[X, Q^{-1}(X)] := \{Y \in \mathcal{L}(W)|\ X \subseteq Y \subseteq Q^{-1}(X)\}$ is an interval of the lattice $\mathcal{L}(W)$ of all subspaces of W. Its length is

$$\dim(Q^{-1}(X)) - \dim(X) \quad = \quad \dim(\ker Q) \quad =: \quad \kappa_1. \tag{17}$$

Since $Q_2 :\ \mathrm{im}(Q) \to\ \mathrm{im}(Q)$ and $\dim(\mathrm{im}Q) = \overline{n} - \kappa_1$ it follows from (16) and (17) that

$$|\mathcal{L}(Q)| \quad\leq\quad |\mathcal{L}(Q_2)| \cdot G(\kappa_1, K) \quad\leq\quad G(\overline{n} - \kappa_1, K) \cdot G(\kappa_1, K). \tag{18}$$

Iterating this idea, observe that $\ker(Q_2) = \ker(Q) \cap \mathrm{im}(Q)$, hence $\kappa_2 := \dim(\ker Q_2) \leq \kappa_1$. Putting $Q_3 := Q_2 \restriction\ \mathrm{im}(Q_2)$ one deduces, as above,

$$|\mathcal{L}(Q_2)| \quad\leq\quad |\mathcal{L}(Q_3)| \cdot G(\kappa_2, K),$$

which, when substituted into (18), yields

$$|\mathcal{L}(Q)| \quad\leq\quad |\mathcal{L}(Q_3)| \cdot G(\kappa_2, K) \cdot G(\kappa_1, K).$$

By induction and because of $|\mathcal{L}(Q_{m+1})| = 1$, one gets

$$|\mathcal{L}(Q)| \quad\leq\quad G(\kappa_m, K) \cdots G(\kappa_2, K) \cdot G(\kappa_1, K),$$

where $\kappa_m \leq \kappa_{m-1} \leq \cdots \leq \kappa_2 \leq \kappa_1$ are defined in the obvious way. Therefore

$$|\mathcal{L}(Q)| \quad\leq\quad G(\dim(\ker Q), K)^m. \tag{19}$$

We are interested, for fixed $\sigma \in S_n$, in the case $K = GF(2), Q = T_1 + I,\ W = V_1,\ \overline{n} = n_1,\ m = \mu_1$. To fix ideas suppose that $(2, 5, 7, 9)$ is one of the cycles of σ. It gives rise to exactly one nonzero $v \in V$ with $T_\sigma(v) = v$; namely $v := e_2 + e_5 + e_7 + e_9$. Therefore $Q(v) = 0$. Thus, clearly $\dim(\ker Q) = r$. See (15) for the relation between r and n_1. Claim (a) now follows from (18) in view of $\mathcal{L}(T_1) = \mathcal{L}(T_1 + I)$. Claim (b) follows from (19). □

Notice that more than $\lfloor n/2 \rfloor! 2^n$ permutations $\sigma \in S_n$ have $T_\sigma = T_1$ or, what amounts to the same, $n_1(\sigma) = n$. This is most easily seen when $n = 2^{\alpha_1}$ happens to be a power of 2. Then even $(n-1)!$ permutations $\sigma \in S_n$ have $n_1(\sigma) = n$, namely by (15) all the n-cycles.

In what follows $r = r(\sigma),\ n_1 = n_1(\sigma)$, and log is the logarithm to base 2. Putting
$\mathcal{D}_1 := \{\sigma \in S_n\ | n_1 \leq n - 6 \log n\}$,
$\mathcal{D}_2 := \{\sigma \in S_n \setminus \mathcal{D}_1\ | 1 \leq r \leq 8 \log n_1\}$,
$\mathcal{D}_3 := \{\sigma \in S_n \setminus \mathcal{D}_1\ | 8 \log n_1 < r < n_1 - 8 \log n_1\}$,
$\mathcal{D}_4 := \{\sigma \in S_n \setminus \mathcal{D}_1\ | n_1 - 8 \log n_1 \leq r \leq n - 1\}$,
it is clear that $S_n - \{id\}$ is the disjoint union of the sets $\mathcal{D}_i$ $(1 \leq i \leq 4)$. The remainder of the article essentially amounts to giving upper bounds for each of the four sums $\sum_{\sigma \in \mathcal{D}_i} |\mathcal{L}(T_\sigma)|$. For $i = 4$ a lower bound will be needed as well.

Lemma 4.

$$\sum_{\sigma \in \mathcal{D}_1} |\mathcal{L}(T_\sigma)| = O(2^{(n^2/4) - n \log n}), \tag{20}$$

$$\sum_{\sigma \in \mathcal{D}_2} |\mathcal{L}(T_\sigma)| = O(2^{17 n \log^2 n}), \tag{21}$$

$$\sum_{\sigma \in \mathcal{D}_3} |\mathcal{L}(T_\sigma)| = O(2^{(n^2/4) - n \log n}). \tag{22}$$

Proof. Without always mentioning it, Lemma 1 will be used throughout the proof. As to (20), fix n and consider the maximum of the function

$$\frac{x^2}{4} + \frac{(n-x)^2}{8} + (0.3)n \quad (0 \le x \le n - 6\log n).$$

Since for big enough n this maximum is obtained at $x = n - 6\log n$, it follows from Lemma 2 (and Lemma 1) that for all $\sigma \in \mathcal{D}_1$

$$|\mathcal{L}(T_\sigma)| = O(2^{\frac{n_1^2}{4}+\frac{(n-n_1)^2}{8}+(0.3)n}) = O(2^{\frac{(n-6\log n)^2}{4}+\frac{(6\log n)^2}{8}+(0.3)n}) = O(2^{\frac{n^2}{4}-2n\log n}),$$

which, in view of $|\mathcal{D}_1| \le n! \le n^n = 2^{n\log n}$, yields

$$\sum_{\sigma\in\mathcal{D}_1} |\mathcal{L}(T_\sigma)| \quad = \quad 2^{n\log n}\cdot O(2^{\frac{n^2}{4}-2n\log n}) \quad = \quad O(2^{\frac{n^2}{4}-n\log n}).$$

As to (21), from $r \le 8\log n_1 \le 8\log n$ and $\mu_1 \le n$ and Lemma 3(b) one deduces

$$|\mathcal{L}(T_1)| \quad \le \quad G(8\log n, 2)^n \ \le \ \left(8\cdot 2^{(8\log n)^2/4}\right)^n \quad = \ O(2^{16n\log^2 n\ +\ 3n}).$$

Since $\sigma \in \mathcal{D}_2$ implies $\sigma \notin \mathcal{D}_1$, whence $n_1 > n - 6\log n$, Lemma 2 yields

$$\sum_{\sigma\in\mathcal{D}_2} |\mathcal{L}(T_\sigma)| \quad = \quad 2^{n\log n}\cdot O(2^{16n\log^2 n+\frac{36\log^2 n}{8}+3.3n}) \quad = \quad O(2^{17n\log^2 n}).$$

As to (22), for all $\sigma \in \mathcal{D}_3$ one derives from Lemma 3(a) that

$$\begin{aligned} |\mathcal{L}(T_1)| \quad &\le \quad G(r,2)\cdot G(n_1-r,2) \quad = \quad O(2^{\frac{r^2}{4}+\frac{(n_1-r)^2}{4}}) \\ &= \quad O(2^{\frac{(8\log n_1)^2}{4}+\frac{(n_1-8\log n_1)^2}{4}}) \quad = \quad O(2^{\frac{n_1^2}{4}-3n_1\log n_1}), \end{aligned}$$

so by Lemma 2

$$|\mathcal{L}(T_\sigma)| = O(2^{\frac{n_1^2}{4}+\frac{(n-n_1)^2}{8}-3n_1\log n_1+(0.3)n}) = O(2^{\frac{n^2}{4}-3n_1\log n_1+(0.3)n}) = O(2^{\frac{n^2}{4}-2n\log n}).$$

Here the last equality holds since $n_1 > n - 6\log n$. As previously, one now argues that

$$\sum_{\sigma\in\mathcal{D}_3} |\mathcal{L}(T_\sigma)| \quad = \quad 2^{n\log n}\cdot O(2^{\frac{n^2}{4}-2n\log n}) \quad = \quad O(2^{\frac{n^2}{4}-n\log n}). \qquad \square$$

The asymptotic behavior of $b(n)$ will depend on the size of

$$Z(n) \ := \ \sum_{\sigma\in\mathcal{D}_4} |\mathcal{L}(T_\sigma)|.$$

Lemma 4 guarantees that the sum of the other $|\mathcal{L}(T_\sigma)|$ is negligible in comparison. By Lemmata 1 and 4 it would suffice to show that $Z(n) = o(2^{n^2/4})$ in order to prove (1). But we strive for more than (1). This requires a *sharper upper* bound for $Z(n)$ (section 4), as well as a *lower* bound for $Z(n)$ (section 3).

3. A lower bound for $Z(n)$. Consider a transposition $\tau \in S_n$. As seen in the introduction, $\mathcal{L}(T_\tau)$ has size at least $G(n-1,2)$. Here is the precise value:

$$\text{If } r(\tau) = n-1, \text{ then } |\mathcal{L}(T_\tau)| = 2G(n-1,2) - G(n-2,2). \tag{23}$$

To see (23) consider without loss of generality the transposition $\tau = (1,2)$. We claim that

$$\mathcal{L}(T_{(1,2)}) = \{U \in \mathcal{L}(V) |\ \langle e_1 + e_2 \rangle \subseteq U \text{ or } U \subseteq \langle e_1 + e_2 \rangle^\perp\}. \tag{24}$$

To see (24), let $U \in \mathcal{L}(T_{(1,2)})$ be such that $e_1 + e_2 \notin U$. We have to show that $U \subseteq \langle e_1 + e_2 \rangle^\perp$. Assume to the contrary some $x = \sum_{i=1}^n \lambda_i e_i$ in U has scalar product $(e_1+e_2) \cdot x \neq 0$. Then $x = e_1 + \sum_{i=3}^n \lambda_i e_i$ or $x = e_2 + \sum_{i=3}^n \lambda_i e_i$, say the former. From $T_{(1,2)}(x) = e_2 + \sum_{i=3}^n \lambda_i e_i$ being in U we get the contradiction $e_1 + e_2 = x + T_{(1,2)}(x) \in U$. This establishes one inclusion in (24). The reverse inclusion is similar and left to the reader.

By (24), $\mathcal{L}(T_{(1,2)})$ is the union of the $G(n-1,2)$-element interval sublattices $[\langle e_1 + e_2 \rangle, V]$ and $[0, \langle e_1 + e_2 \rangle^\perp]$, whose intersection is the $G(n-2,2)$-element interval sublattice $[\langle e_1 + e_2 \rangle, \langle e_1 + e_2 \rangle^\perp]$. This gives (23).

We now double the lower bound in (4). More precisely, because $G(n-2,2) = o(G(n-1,2))$ it follows from (23) and Lemma 1 that

$$\sum_{r(\sigma)=n-1} |\mathcal{L}(T_\sigma)| \quad \geq \quad \binom{n}{2} \cdot 2 \cdot 7.3719 \cdot 2^{\frac{(n-1)^2}{4}} \qquad (n \text{ large}). \tag{25}$$

Because $r(\sigma) = n-1$ implies $\sigma \in \mathcal{D}_4$, the right-hand side of (25) is also a lower bound for $Z(n)$.

4. An upper bound for $Z(n)$. From Lemma 1 and the proof of (25) it follows at once that upon transition from 7.3719 to 7.37197 one has

$$\sum_{r(\sigma)=n-1} |\mathcal{L}(T_\sigma)| \quad \leq \quad \binom{n}{2} \cdot 2 \cdot 7.37197 \cdot 2^{\frac{(n-1)^2}{4}} \qquad (n \text{ large}). \tag{26}$$

In order to prove that

$$\sum_{\sigma \in \mathcal{D}_4} |\mathcal{L}(T_\sigma)| \quad \leq \quad \binom{n}{2} \cdot 2 \cdot 7.37198 \cdot 2^{\frac{(n-1)^2}{4}} \qquad (n \text{ large}), \tag{27}$$

put

$$\mathcal{D} := \{\sigma \in S_n |\ n_1(\sigma) > n - 6\log n \text{ and } n - 14\log n \leq r(\sigma) \leq n-1\}.$$

All $\sigma \in \mathcal{D}_4$ satisfy $n_1(\sigma) > n - 6\log n$, as well as

$$r \quad \geq \quad n_1 - 8\log n_1 \quad > \quad (n - 6\log n) - 8\log n \quad = \quad n - 14\log n,$$

so $\mathcal{D}_4 \subseteq \mathcal{D}$. In view of (26) it thus suffices to show

$$\sum_{\sigma \in \mathcal{D}, r(\sigma) \leq n-2} |\mathcal{L}(T_\sigma)| \quad = \quad o(2^{\frac{(n-1)^2}{4}}) \quad = \quad o(2^{\frac{n^2}{4} - \frac{n}{2}}). \tag{28}$$

Fix $\sigma \in \mathcal{D}$ with $r(\sigma) \leq n-2$. Consider T_σ and the associated T_1. Putting $n_1 := n_1(\sigma)$ and $r := r(\sigma)$, Lemma 3(a) yields

$$\begin{aligned}|\mathcal{L}(T_1)| &\leq G(r,2) \cdot G(n_1 - r, 2)\\ &= O(2^{\frac{r^2}{4}+\frac{(n_1-r)^2}{4}}) = O(2^{\frac{(n-2)^2}{4}+\frac{2^2}{4}}) = O(2^{\frac{n^2}{4}-n}).\end{aligned}$$

From $n_1 > n - 6\log n$ and Lemma 2 one concludes that

$$|\mathcal{L}(T_\sigma)| \quad \leq \quad 2^{\frac{(n-n_1)^2}{8}+(0.3)n} \cdot O(2^{\frac{n^2}{4}-n}) \quad = \quad O(2^{\frac{n^2}{4}-(0.7)n+\frac{36\log^2 n}{8}}).$$

How many elements has $\mathcal{D}$ at most? We claim that $\mathcal{D}$ is contained in the class $\mathcal{D}'$ of all $\sigma \in S_n$, which have at least $n-28\log n$ cycles of length 1. Indeed, if $\sigma \in \mathcal{D}$ had less than $n - 28\log n$ of them, then σ had less than $(n-28\log n) + 14\log n = n - 14\log n$ cycles altogether, contradicting the definition of $\mathcal{D}$. Hence

$$|\mathcal{D}| \quad \leq \quad |\mathcal{D}'| \quad \leq \quad \binom{n}{n-28\log n}[(28\log n)!] \quad \leq \quad n^{28\log n},$$

which implies

$$\sum_{\sigma\in\mathcal{D},\ r(\sigma)\leq n-2} |\mathcal{L}(T_\sigma)| \quad = \quad 2^{28\log^2 n} \cdot O(2^{\frac{n^2}{4}-(0.7)n+\frac{36\log^2 n}{8}}) \quad = \quad o(2^{\frac{n^2}{4}-\frac{n}{2}}).$$

This proves (28) and whence (27).

5. The main theorem.

5. The main theorem. We are now in a position to prove the following.

THEOREM. *For all sufficiently large n one has*

$$\left(1+2^{-\frac{n}{2}+2\log n+0.2499}\right)\frac{G(n,2)}{n!} \quad \leq \quad b(n) \quad \leq \quad \left(1+2^{-\frac{n}{2}+2\log n+0.2501}\right)\frac{G(n,2)}{n!}.$$

Proof. By Lemma 1 one has $G(n,2) \leq 7.3720 \cdot 2^{n^2/4}$ for all large enough n. Together with (2) and (25) this implies that for large enough n

$$\begin{aligned}b(n) = \frac{G(n,2)}{n!}&\left(1+\frac{1}{G(n,2)}\sum_{r(\sigma)\leq n-1}|\mathcal{L}(T_\sigma)|\right)\\ &\geq \quad \frac{G(n,2)}{n!}\left(1+\binom{n}{2}2^{-\frac{n}{2}+1.25}\cdot\frac{7.3719}{7.3720}\right)\\ &\geq \quad \frac{G(n,2)}{n!}\left(1+2^{-\frac{n}{2}+2\log n+0.2499}\right).\end{aligned}$$

The last inequality holds because

$$\binom{n}{2}\cdot\frac{7.3719}{7.3720} \;=\; \frac{n^2}{2}\left(1-\frac{1}{n}\right)\cdot\frac{7.3719}{7.3720} \;\geq\; \frac{n^2}{2}\cdot 2^{-0.00001}\cdot 2^{-0.00002} \;=\; 2^{2\log n - 1.00003}$$

for large n. From Lemma 4 and (27) we see that

$$(29) \qquad \sum_{r(\sigma)\leq n-1}|\mathcal{L}(T_\sigma)| \quad \leq \quad \binom{n}{2}\cdot 2\cdot 7.3720\cdot 2^{\frac{(n-1)^2}{4}} \qquad (n \text{ large}).$$

By Lemma 1 one has $G(n,2) \geq 7.3719 \cdot 2^{n^2/4}$ for all large enough n, so (29) yields

$$\begin{aligned} b(n) = \frac{G(n,2)}{n!} &\left(1 + \frac{1}{G(n,2)} \sum_{r(\sigma)\leq n-1} |\mathcal{L}(T_\sigma)|\right) \\ \leq \quad \frac{G(n,2)}{n!} &\left(1 + \binom{n}{2} 2^{-\frac{n}{2}+1.25} \cdot \frac{7.3720}{7.3719}\right) \\ \leq \quad \frac{G(n,2)}{n!} &\left(1 + 2^{-\frac{n}{2}+2\log n + 0.2501}\right). \qquad \square \end{aligned}$$

It should be clear from the proof that the exponents 0.2499 and 0.2501 in the theorem *cannot* be replaced by $0.25 \pm \epsilon$. However, equally clear, $0.25 \pm \epsilon$ can be introduced if one distinguishes between even and odd integers. It is also obvious that the theorem implies (1). In turn, (1) implies that the fraction $\beta(n)$ of n-codes X with nontrivial automorphism group $\mathrm{Aut}(X) := \{\sigma \in S_n |\ X_\sigma = X\}$ goes to 0 for $n \to \infty$. Namely, the total number $b(n)$ of equivalence classes satisfies

$$(30) \quad b(n) \quad \geq \quad \frac{\beta(n)G(n,2)}{n!/2} + \frac{(1-\beta(n))G(n,2)}{n!} \quad = \quad \frac{(1+\beta(n))G(n,2)}{n!}.$$

By (1) this forces $\beta(n) \to 0$ as $n \to \infty$. Notice that there is no quick argument why, conversely, (30) together with $\beta(n) \to 0$ should imply (1). This relates to results in [LPR]; see [W2] for details. A year after [W2] the mistake in [W1] was also fixed in [H]; in fact Hou extends formula (1) to prime powers $q > 2$.

REFERENCES

[A] M. Aigner, *Combinatorial Theory*, Springer-Verlag, New York, 1979.

[BF] L. Brickman and P. A. Fillmore, *The invariant subspace lattice of a linear transformation*, Canad. J. Math., 19 (1967), pp. 810–822.

[H] X. D. Hou, *On the asymptotic number of non-equivalent q-ary linear codes*, J. Combin. Theory Ser. A, 112 (2005), pp. 337–346.

[L] R. F. Lax, *On the character of S_n acting on subspaces of* $\mathbb{F}_q^n$, Finite Fields Appl., 10 (2004), pp. 315–322.

[LPR] H. Lefmann, K. T. Phelps, and V. Rödl, *Rigid linear binary codes*, J. Combin. Theory Ser. A, 63 (1993), pp. 110–128.

[O] J. G. Oxley, *Matroid Theory*, Oxford University Press, New York, 1997.

[W1] M. Wild, *The asymptotic number of inequivalent binary codes and nonisomorphic binary matroids*, Finite Fields Appl., 6 (2000), pp. 192–202.

[W2] M. Wild, *The asymptotic number of binary codes and binary matroids*. This is a previous version of the present article with connections to [LPR]. Also available online at http://arxiv.org/abs/cs.IT/0408011.

SIAM J. DISCRETE MATH.
Vol. 19, No. 3, pp. 700–718

GENERALIZED ALON–BOPPANA THEOREMS AND ERROR-CORRECTING CODES*

JOEL FRIEDMAN† AND JEAN-PIERRE TILLICH‡

Abstract. In this paper we describe several theorems that give lower bounds on the second eigenvalue of any quotient of a given size of a fixed graph, G. These theorems generalize Alon–Boppana-type theorems, where G is a regular (infinite) tree.

When G is a hypercube, our theorems give minimum distance upper bounds on linear binary codes of a given size and information rate. Our bounds at best equal the current best bounds for codes and apply only to linear codes. However, it is of interest to note that (1) one very simple Alon–Boppana argument yields nontrivial code bound, and (2) our Alon–Boppana argument that equals a current best bound for codes has some hope of improvement.

We also improve the bound in sharpest known Alon–Boppana theorem (i.e., when G is a regular tree).

Key words. eigenvalues, graphs, error-correcting codes, Alon–Boppana, expanders, Faber–Krahn

AMS subject classifications. 68R10, 94B65

DOI. 10.1137/S0895480102408353

1. Introduction.

The goal of this paper is to draw a connection between the "Alon–Boppana" bound, in the theory of expanders or graph eigenvalues, and asymptotic upper bounds for the minimum distance of an error-correcting code of a given rate.

Recall that the Alon–Boppana bound is a lower bound on the second eigenvalue of finite d-regular graphs. In its basic form it says that the second largest eigenvalue of a d-regular graph is greater than $2\sqrt{d-1} - o(1)$ as the number of vertices goes to infinity. In this paper we show that the Alon–Boppana bound can be generalized to finite quotients of a large class of graphs, H; in the original Alon–Boppana setting, H is the d-regular infinite tree, which covers any (connected) d-regular graph. See [Fri03] for such results when H is infinite and fixed.

The connection with upper bounds on the minimum distance of a binary linear code is that the minimum distance of a binary linear code C can be expressed as a certain decreasing function of the second largest eigenvalue of a certain regular graph associated to C (this graph is generally called the *coset graph* of $C^{\perp}$; see section 5). In other words any lower bound on the second eigenvalue of this graph translates into an upper bound on the minimum distance of the code. If we use the aforementioned Alon–Boppana bound directly, then we only obtain a very weak upper bound on the minimum distance of the code.

However, when we know more about the geometry of the graph—for instance, lower bounds on the number of cycles of a given length—then the Alon–Boppana lower bound can be strengthened considerably. We derive several lower bounds using

*Received by the editors May 27, 2002; accepted for publication (in revised form) April 15, 2005; published electronically November 23, 2005.

http://www.siam.org/journals/sidma/19-3/40835.html

†Departments of Computer Science and Mathematics, University of British Columbia, Vancouver, BC V6T 1Z4 (V6T 1Z2 for Mathematics), Canada (jf@cs.ubc.ca). This author's research was supported in part by an NSERC grant.

‡Inria Projet Codes, Domaine de Voluceau BP.105, 78153, France (jean-pierre.tillich@inria.fr). This author's research was supported in part by France Telecom.

different techniques. The first one is derived through lower bounds on the number of cycles of a given length, the second through comparison with Dirichlet eigenvalues. There is, however, a common underlying idea, namely, the notion of a covering graph (see section 3). In both cases, the relevant quantities (either the number of cycles or the Dirichlet eigenvalues) are bounded by the corresponding quantities of a cover graph. The crux of this approach is that the cover graph may have a simple structure (for instance, for the coset graph we may choose a Boolean hypercube), which enables us to estimate these quantities directly.

The second technique, when applied to the graph associated to the coset graph of a binary linear code, yields the first MRRW bound [MRRW77] in coding theory, which is the best known upper bound on the minimum distance for low-rate codes. This bound was originally obtained with the "linear programming" approach. Although our approach has elements in common with the classical "linear programming" approach, we believe our approach is easier to use and suggests more geometrically visualizable questions on the Boolean hypercube. This is because a simple "Alon–Boppana" argument easily gives an interesting coding bound (see section 5), and we don't know of an analogous argument based on the linear programming approach. Also, in an attempt to improve the "first MRRW bound" (of [MRRW77], as explained in section 2) there arises a geometric question about what is the correct analogue for the hypercube of the classical Faber–Krahn inequality for domains in $\mathbb{R}^n$ (see, e.g., [Fab23, Kra25, Cha84, Fri93]); if this analogue is "asymptotically different" (see section 10), which is presently conceivable, then the first MRRW bound will be improved. We must admit, however, that at present we cannot improve but only duplicate the first MRRW bound with our methods; furthermore, it is quite conceivable that any theorem obtained with our methods could be translated into a proof based only on the linear programming approach (it would be interesting to know if this were really true). But we reiterate that even if our approach is, in a sense, subsumed by the linear programming approach, the setting and geometric pictures suggested by our method seems to be easier to work with. Moreover, we also show how to obtain the linear programming bounds dealing only with the Hamming space through our approach, by slightly changing one of our Alon–Boppana bounds (see section 10).

The consequences for the classical Alon–Boppana theorem (i.e., for the second eigenvalue of a d-regular graph) in this paper is that we improve the best Alon–Boppana bound (of Friedman and Kahale; see [Fri93]) in the second-order term by essentially a factor of 4 (see section 9). This is done by generalizing the known Alon–Boppana bound techniques to give coding bounds, and realizing that the first MRRW bound improves this bound, in a sense, by a factor of 2. It is not hard to see where this factor of 2 can be recovered—by "projecting out the constants" (see sections 9 and 10). However, Kahale's method (see [Kah93]) also "projects out the constants," and our improvement to classical Alon–Boppana can also be obtained by a minor modification of Kahale's proof.

2. A basic fact for obtaining Alon–Boppana bounds.

Let us first introduce some general notation concerning eigenvalues of (adjacency matrices of) graphs. Let G be a graph with $|V_G| = n$ and adjacency matrix A_G. Recall that A_G is an $n \times n$ symmetric matrix, with entries a_{uv} indexed by the vertices of the graph, and $a_{uv} = 1$ iff u and v are adjacent in G, and $a_{uv} = 0$ otherwise. Since A_G is symmetric, it can be diagonalized with an orthonormal basis. Then we can write

$$\lambda_1(G) \geq \lambda_2(G) \geq \cdots \geq \lambda_n(G)$$

for the eigenvalues of G's adjacency matrix (written with their multiplicity). We denote by $e_1, e_2, \dots, e_n$ the corresponding (orthonormal) basis of eigenvectors. We write $\rho_i = \rho_i(G)$ for the ith largest value that occurs among the $|\lambda_i|$; for example, the Perron–Frobenius theorem implies that $\rho_1 = \lambda_1$ and thus

$$\rho_2 = \rho_2(G) = \max(\lambda_2, -\lambda_n).$$

Estimating λ_2 is of interest in studying expansion; however, some techniques estimate only ρ_2 (and higher ρ_i).

The Rayleigh principle gives us the following characterization of $\lambda_2(G)$ (it is a straightforward consequence of the fact that $e_1, e_2, \dots, e_n$ is an orthonormal basis):

$$\lambda_2(G) = \max_{f \perp e_1} \frac{(A_G f, f)}{(f, f)}. \tag{2.1}$$

If G is a regular graph, then e_1 can be chosen to be $\frac{1}{\sqrt{n}}\vec{1}$, where $\vec{1}$ is the all-ones vector, and, therefore, by applying the previous equation we obtain following fact.

FACT 2.1. *If G is a regular graph and $f \in \mathbb{R}^n$ is orthogonal to $\vec{1}$, then*

$$\lambda_2(G) \geq \frac{(A_G f, f)}{(f, f)}. \tag{2.2}$$

This inequality is the key to obtaining lower bounds on $\lambda_2(G)$: by choosing f appropriately we can relate $\lambda_2(G)$ to other quantities of the graph. Notice that we can also apply the Rayleigh principle to A_G^l (or even sometimes to a well-chosen polynomial applied to A_G); this yields for $f \perp \vec{1}$ and any positive odd integer l,

$$\lambda_2(G)^l \geq \frac{(A_G^l f, f)}{(f, f)}, \tag{2.3}$$

and in general for any positive integer l,

$$\rho_2(G)^l \geq \frac{(A_G^l f, f)}{(f, f)}. \tag{2.4}$$

In what follows we are going to apply these simple facts to several different choices of f. For all these choices we are going to control the term $\frac{(A_G^l f, f)}{(f, f)}$ that appears on the right-hand side through the notion of a cover graph.

3. Graphs and covers.

In this section we review the definition of graph covers. Until section 11 we assume all graphs are *simple*, i.e., having no multiple edges or self-loops; this simplifies the discussion and notation. In section 11 we give the definitions needed for general graphs; all theorems immediately carry over to general graphs.

By a *simple graph* we mean a graph with no multiple edges or self-loops; so we may think of a simple graph, G, as a pair (V_G, E_G), where E_G is a subset of the set of unordered pairs of V_G. Until section 11 we understand a *graph* to mean a simple graph.

A *morphism* $\pi\colon H \to G$ of graphs is a map from V_H to V_G such that the natural map from E_H onto pairs in V_G has its image in E_G. π thus gives rise to a map from E_H to E_G which we also denote by π, assuming no confusion will arise.

A morphism $\pi\colon H \to G$ is called a *covering map* if for every edge $e = \{u, v\}$ of G and every $u' \in V_H$ with $\pi(u') = u$ there is a unique $v' \in \pi^{-1}(v)$ such that $\{u', v'\}$ is an edge in E_H. We also say that in this case H is a *cover* of G.

Example 3.1. Let G be any finite graph. Then G has a *universal cover*, $\pi\colon T \to G$, in that for any covering map $\nu\colon K \to G$ there is a covering map[1] $\mu\colon T \to K$ such that $\pi = \nu \circ \mu$. T is a tree. If G is d-regular, i.e., each row and column of A_G sums to d, then T is a d-regular tree (and any two d-regular trees are isomorphic).

Example 3.2. Let G be a connected Cayley graph on $(\mathbb{F}_2)^k$ of degree n with generators $c_1, \dots, c_n$. This is a graph where we connect any $x \in (\mathbb{F}_2)^k$ to $x + c_i$ for $i \in \{1, 2, \dots, n\}$. Let $\mathbb{B}^n$ be the Boolean n-hypercube, i.e., the Cayley graph on $(\mathbb{F}_2)^n$ with generators $e_1, \dots, e_n$, where e_i is the ith standard basis vector, i.e., e_i is 0 on each coordinate except the ith, where it is 1. Consider the map $\pi_{\mathrm{lin}}\colon (\mathbb{F}_2)^n \to (\mathbb{F}_2)^k$ which takes e_i (as above) to c_i and is extended by linearity. Then π_{lin} induces a covering map $\pi\colon \mathbb{B}^n \to G$.

4. Coding theory. A *code* of *length* n is a subset $C \subset (\mathbb{F}_2)^n$, where $\mathbb{F}_2 = \{0, 1\}$ is the field with two elements. C is *linear* if it is a subspace of the vector space $(\mathbb{F}_2)^n$. We endow $(\mathbb{F}_2)^n$ with the Hamming distance, i.e., for $x, y \in (\mathbb{F}_2)^n$, $d(x, y)$ is the number of coordinates on which x and y differ. The minimum distance of a code, C, is

$$d_{\min}(C) = \min\{d(x, y) \mid x, y \in C,\ x \neq y\},$$

and its *normalized minimum distance* is

$$\delta(C) = d_{\min}(C)/n.$$

The *information rate* of a code is

$$R(C) = \frac{\log_2 |C|}{n}.$$

If C is a linear code, then this is just $(\dim C)/n$.

Let $\delta_{\max}$ be the function

$$\delta_{\max}(R) = \overline{\lim_{n\to\infty}} \max\{\delta(C) \mid R(C) \geq R,\ C \subset (\mathbb{F}_2)^n\}$$

and

$$R_{\max}(\delta) = \overline{\lim_{n\to\infty}} \max\{R(C) \mid \delta(C) \geq \delta,\ C \subset (\mathbb{F}_2)^n\}.$$

We are interested in estimating these functions.

Estimating $\delta_{\max}$ is essentially the same as estimating $R_{\max}$, but a bit of care is required to make this precise.

PROPOSITION 4.1. *Let $\delta_{\max}(\alpha) \leq f(\alpha)$ for a continuous, strictly decreasing function, f, defined on an open interval. Then (f^{-1} is defined on the image of f and) $R_{\max}(\delta) \leq f^{-1}(\delta)$.*

Proof. This is an easy (but mildly annoying) technicality; see section 14. □

We now state some classical bounds.

THEOREM 4.2. *$R_{\max}(\delta) \geq 1 - h(\delta)$, where*

$$h(\theta) = -\theta \log_2 \theta - (1 - \theta) \log_2(1 - \theta).$$

[1] This covering map, μ, is uniquely defined if one works with "base-pointed graphs," i.e., graphs with a distinguished vertex.

Proof. See the asymptotic Gilbert–Varshamov bound in [vL99]. □

The best upper bound on $R_{\max}$ is given by the following theorem.

THEOREM 4.3.

$$R_{\max}(\delta) \leq \min_{u\in[0,1-2\delta]} b(u,\delta), \tag{4.1}$$

where

$$b(u,\delta) = 1 + g(u^2) - g(u^2 + 2\delta u + 2\delta)$$

with

$$g(x) = h\left(\frac{1}{2} - \frac{\sqrt{1-x}}{2}\right).$$

For $\delta \geq 0.273$ *this bound is the same as*

$$R_{\max}(\delta) \leq b(1-2\delta,\delta) = h\left(1/2 - \sqrt{\delta(1-\delta)}\right). \tag{4.2}$$

Proof. See [MRRW77] (or [MS77v1, MS77v2] for the latter half of the theorem). □

The inequality (4.1) is known as the "second MRRW" bound; (4.2) is known as the "first MRRW" bound.

COROLLARY 4.4. *For small* α *we have*

$$\frac{1}{2} - (1+o(1))\sqrt{\frac{\alpha}{2\log_2 e}} \leq \delta_{\max}(\alpha) \leq \frac{1}{2} - (1+o(1))\sqrt{\frac{\alpha}{\log_2(1/\alpha)}}.$$

5. Codes and eigenvalues. In this section we recall how a graph can be associated to a linear code in such a way that the eigenvalues of the graph are in relationship with the codeword weights.

Let $C \subset (\mathbb{F}_2)^n$ be a linear code with basis $r_1,\dots,r_k$. We form the generator matrix, M, over $\mathbb{F}_2$, whose rows are the r_i's; so M is a $k\times n$ matrix. Its columns, $c_1,\dots,c_n$, can each be viewed as an element of $(\mathbb{F}_2)^k$.

Let G be the Cayley graph on $(\mathbb{F}_2)^k$ with generators $c_1,\dots,c_n$.[2] Apparently G may depend on the choice of the basis $r_1,r_2,\dots,r_k$. It turns out that G depends only on C. This can be seen by bringing in the dual code $C^\perp$ of C, that is,

$$C^\perp = \{x \in (\mathbb{F}_2)^n \mid x\cdot c = 0 \quad \forall c \in C\}.$$

Consider the graph with vertices the cosets $x + C^\perp$, and two cosets being linked by an edge iff they are at Hamming distance 1. We claim that the Cayley graph defined before and this new graph are isomorphic, the isomorphism being given by the map $\pi : x + C^\perp \to Mx$. Indeed, let two cosets $x + C^\perp$ and $y + C^\perp$ be linked by an edge. This means that there exists $c \in C^\perp$ and $i \in \{1,\dots,n\}$ such that $x = y+c+e_i$ (where e_i is the ith standard basis vector of ${\mathbb{F}_2}^n$, i.e., e_i is 0 on each coordinate except the ith, where it is 1); this implies that $Mx = My + c_i$. On the other hand if $Mx = My + c_i$, then necessarily x and $y + e_i$ differ only by an element of $C^\perp$.

[2] We shall assume (until section 11) that no c_i's vanish and the c_i's are all distinct; if not, then G will have self-loops and/or multiple edges, and we technically need section 11 before we can apply our theory.

We say that this graph is the *coset graph* of $C^{\perp}$ or of the code,[3] C. The following is a well-known folk theorem (see [DS91] and the references therein).

THEOREM 5.1. *Let $\lambda_1 \geq \lambda_2 \geq \cdots$ be the eigenvalues of the adjacency matrix of the coset graph of $C^{\perp}$ arranged in nonincreasing order. Then $\lambda_1 = n$ and $\lambda_2 = n - 2d_{min}(C)$. Moreover, the weights (i.e., distances to the zero code word) appearing in C are just the $(n - \lambda_i)/2$ as i ranges from 1 to 2^k.*

6. A simple generalized Alon–Boppana theorem. In this section we give a very simple but rather weak generalized Alon–Boppana theorem and discuss its implications. Let G be a d-regular graph. We use the approach outlined in section 2 to obtain a lower bound on $\lambda_2(G)$ and $\rho_2(G)$ and we choose $f = \chi_u - \chi_v$ where χ denotes the characteristic function in (2.4). Notice that

$$\left(A_G^l \chi_u, \chi_u\right) = N_l(u), \qquad \left(A_G^l \chi_v, \chi_v\right) = N_l(v),$$

where $N_l(v)$ denotes the number of walks of length l from v to itself. Moreover, if u and v are at distance greater than $l \geq 0$, then

$$\left(A_G^l \chi_u, \chi_v\right) = \left(A_G^l \chi_v, \chi_u\right) = 0.$$

Hence

$$\left(A_G^l f, f\right) = N_l(u) + N_l(v).$$

Let $N_l = N_l(G)$ denote the minimum of $N_l(v)$ ranging over all vertices v of the graph. Of course, $(f, f) = 2$, and so

$$\frac{\left(A_G^l f, f\right)}{(f, f)} \geq N_l(G).$$

By using (2.4) we now obtain

$$\rho(A_G) \geq \left(N_l\right)^{1/l}.$$

The right-hand term can be estimated through a cover H of G for which the calculation of $N_l(H)$ might be much simpler. Indeed the following is clear.

FACT 6.1. *If $\pi \colon H \to G$ is a cover, then any H cycle about a vertex, v, gives rise to a unique G cycle about $\pi(v)$. Hence for any positive integer l we have*

$$N_l(G) \geq N_l(H).$$

In other words we have proved the following.

THEOREM 6.2. *Let G be a d-regular graph that contains two vertices of distance greater than l and let H be a cover of G. Then*

$$\rho(A_G) \geq \left(N_l(H)\right)^{1/l};$$

furthermore, if l is odd, then the above equation holds with ρ replaced by λ_2.

[3] This is the graph of cosets of the hypercube modulo $C^{\perp}$, or of $C^{\perp}$ cosets, but it is the graph of cosets one uses when working with C. Since we do not work with a code, C, and its dual, $C^{\perp}$, simultaneously (in this paper), no confusion will occur in referring to the graph as the coset graph of "the code."

The last statement follows by using (2.3) instead of (2.4).

The above theorem is quite simple. Unfortunately, for some purposes, such as coding theory, we are interested in $\lambda_2(A_G)$ and the cover graph H (which can be chosen to be a boolean hypercube) will be bipartite (i.e., $N_l(H) = 0$ for l odd). So we prove the following variant of the above theorem.

THEOREM 6.3. *Let $\pi\colon H \to G$ be a covering map. Let e_1, e_2 be two edges of distance greater than l (i.e., the distance from any of e_1's endpoints to any of e_2's is greater than l). Then*

$$\rho(A_G) \geq \big(N_l(H) + N_{l-1}(H)\big)^{1/l};$$

furthermore, if l is odd, then the above equation holds with ρ replaced by λ_2.

Proof. Let $e_i = \{u_i, v_i\}$ and set

$$f = \chi_{u_1} + \chi_{v_1} - \chi_{u_2} - \chi_{v_2}.$$

We have that $(A^l\chi_{u_1}, \chi_{v_1})$ is at least $N_{l-1}(H)$, since any walk of length $l-1$ beginning and ending in u_1 yields a walk from u_1 to v_1 with one additional step. Similar reasoning to that in the previous theorem then yields

$$(A^l f, f) \geq 4\big(N_l(H) + N_{l-1}(H)\big)$$

and, of course, $(f, f) = 4$. Similar reasoning as before now yields this theorem. □

We state two corollaries of this simple theorem.

COROLLARY 6.4. *Fix d. Then for any d-regular graph, G, on n vertices, we have $\rho(G) \geq 2\sqrt{d-1} - o(1)$ as $n \to \infty$.*

This follows by taking H to be the universal cover of G (namely, the infinite d-regular tree) and by noticing that any d-regular graph on n vertices has at least two vertices which are at distance $\lfloor \log_{d-1} n \rfloor$. The relevant computation can be found in [LPS88], for instance.

We get stronger bounds with regular graphs which admit a cover which has more closed walks than the d-regular infinite tree, and this is exactly what happens for the coset graph of a code of length n which admits the boolean hypercube as a cover (see Example 3.2).

COROLLARY 6.5. *Let C be a binary linear code of length n of rate $\leq R$. The normalized minimum distance of C, δ, satisfies $\delta \leq f(R)$, where f is a function that satisfies*

$$f(R) = \frac{1}{2} - C\big(1 + o(1)\big)\sqrt{\frac{R}{\log_2 R}}$$

when R tends to 0 with $C = 1/\sqrt{4e}$.

The bound of [MRRW77] yields the same corollary but with $C = 1$. The calculations which leads to this theorem are in section A.

7. Projecting out constants. In this section we introduce a technique that will strengthen essentially all of our Alon–Boppana theorems, including the ones in the previous section and the more refined theorems to come.

In the previous section we created functions, f, for which $(A^l f, f)$ could be bounded; the idea was to concentrate f at a few vertices. Since it is important that f be orthogonal to $\vec{1}$, the all-ones vector, we took f to have as many positive values as negative values, taking the values of different sign to be far apart (a distance greater than l). However, we may alternatively take f to be all positive, provided that we then remove f's component in the direction of $\vec{1}$. This is the same as taking f to be concentrated and positive, subtracting the same (small) negative value at every other vertex.

The idea of choosing an arbitrary f and "projecting out the constant component" will be used repeatedly in this paper. Here is this technique applied to Theorem 6.2.

Theorem 7.1. *Let $\pi\colon H \to G$ be a covering map. Let G be a d-regular graph, and let l be a value such that*

$$N_l(H) \geq d^l/|V_G|.$$

Then

$$\rho(A_G) \geq \left(N_l(H) - \frac{d^l}{|V_G|}\right)^{1/l};$$

furthermore, if l is odd, then the above equation holds with ρ replaced by λ_2.

Proof. Fix any $v \in V_G$ and set $f = \chi_v$. Then $\widetilde{f} = f - \vec{1}/|V_G|$ is orthogonal to $\vec{1}$. We have

$$\left(A^l\widetilde{f}, \widetilde{f}\right) = \left(A^l f, f\right) - \left(A^l\vec{1}/|V_G|, \vec{1}/|V_G|\right) \geq N_l(H) - d^l/|V_G|$$

and

$$\left(\widetilde{f}, \widetilde{f}\right) = (f, f) - (\vec{1}/|V_G|, \vec{1}/|V_G|) \leq 1.$$

The reasoning used at the end of Theorem 6.2 now applies here, and we conclude the theorem. □

We may also obtain the following variant of Theorem 6.3.

Theorem 7.2. *Let $\pi\colon H \to G$ be a covering map with G a d-regular graph, and let l be a value such that*

$$N_l(H) + N_{l-1}(H) \geq 2d^l/|V_G|.$$

Then

$$\rho(A_G) \geq \left(N_l(H) + N_{l-1}(H) - \frac{2d^l}{|V_G|}\right)^{1/l};$$

furthermore, if l is odd, then the above equation holds with ρ replaced by λ_2.

Proof. Fix any edge, $e = \{u, v\}$, and let $f = \chi_u + \chi_v$ and $\widetilde{f} = f - 2\vec{1}/|V_G|$. As before, $\widetilde{f}$ is orthogonal to $\vec{1}$, and we have

$$(A^l\widetilde{f}, \widetilde{f}) = (A^l f, f) - 4(A^l\vec{1}/|V_G|, \vec{1}/|V_G|) \geq 2N_l(H) + 2N_{l-1}(H) - 4d^l/|V_G|$$

and

$$(\widetilde{f}, \widetilde{f}) \leq 2.$$

We argue as before. □

Using this theorem we improve Corollary 6.5 by a factor of 2, as follows; see the appendices for the proof.

Corollary 7.3. *In Corollary 6.5, we may take $C = 1/\sqrt{2e}$.*

8. Eigenfunction pushing techniques. Let us now apply (2.2) to functions of the form $\widetilde{f} = f - c\vec{1}$ (where c is chosen such that $\widetilde{f}$ is orthogonal to $\vec{1}$), where f is a function supported on a subset U of vertices of the graph (this means that f is equal to 0 outside of U). We easily obtain the following.

PROPOSITION 8.1. *Let f be supported on a set, U. Let G be d-regular. Then*

$$\lambda_2(G) \geq \frac{(A_G f, f)}{(f,f)} - \frac{d|U|}{|V_G|}.$$

Proof. Let $\widetilde{f} = f - c\vec{1}$, where $c = (f,\vec{1})/|V_G|$; then $\widetilde{f}$ is orthogonal to $\vec{1}$, and so

$$\lambda_2(G) \geq \frac{(A_G\widetilde{f}, \widetilde{f})}{(\widetilde{f}, \widetilde{f})}. \tag{8.1}$$

Since

$$(f,\vec{1})^2 = (f,\chi_U)^2 \leq (f,f)(\chi_U,\chi_U) = (f,f)|U|,$$

we have

$$(A_G\widetilde{f}, \widetilde{f}) = (A_G f, f) - dc^2|V_G| \leq (A_G f, f) - d(f,f)|U|/|V_G|.$$

Combining this with the fact that $(\widetilde{f}, \widetilde{f}) \leq (f,f)$ (since $\widetilde{f}$ is a projection of f onto the subspace orthogonal to $\vec{1}$) and with (8.1) finishes the proposition. □

To optimize this inequality we have to find for a given subset of vertices U the function f which maximizes the ratio $\frac{(A_G f,f)}{(f,f)}$. This maximum is known as a Dirichlet eigenvalue. We define for a graph G and a subset of vertices $W \subset V_G$,

$$\lambda_{1,\mathrm{Dir}}(W) = \max_{f \in C_0(W)} \frac{(Af,f)}{(f,f)},$$

where we write $C_0(W)$ for those functions supported in W. It is easy to check that the maximum is attained for a nonnegative function (this is a simple consequence of the Perron–Frobenius theorem; see also [Fri93]). The f achieving the above maximum is called the *first Dirichlet eigenfunction* of A; this f is known to satisfy $Af = \lambda f$ for $\lambda = \lambda_{1,\mathrm{Dir}}(W)$ (see [Fri93]).

Then it makes sense to find the subset W of a given size which maximizes this eigenvalue; this leads us to define, for $a > 0$, $\mathrm{FK}_G(a)$, the *Faber–Krahn maximum of size a* as

$$\mathrm{FK}_G(a) = \max_{|W| \leq a} \lambda_{1,\mathrm{Dir}}(W);$$

the W achieving this maximum is the *Faber–Krahn maximizer of size a*.

The nice thing about this quantity is that it has a lower bound in terms of the Faber–Krahn maximum for the same size of a cover graph.

THEOREM 8.2. *Let H be a cover of G. Then*

$$\mathrm{FK}_H(a) \leq \mathrm{FK}_G(a).$$

To prove this fact we need a lemma and a definition. For a covering map $\pi\colon H \to G$ and $f : V_H \to \mathbb{R}$, we define the *push forward*, $\pi_* f$, a function on V_G,whenever H is finite, via

$$(\pi_* f)(v) = \sum_{\pi(w)=v} f(w).$$

LEMMA 8.3. *Let $f \in C(V_H)$ and let $\pi\colon H \to G$ be a covering map. Assume H is finite. If $f \geq 0$ everywhere, then also $\pi_* f \geq 0$. If $A_H f \geq \lambda f$ everywhere, for some real λ, then also $A_G \pi_* f \geq \lambda \pi_* f$. If f is supported in W, then $\pi_* f$ is supported in $\pi(W)$.*

Proof. The first part (the nonnegativity statement) is clear. The second part follows from the fact that π is a local isomorphism. The third part is also clear. □

We are ready now to prove Theorem 8.2.

Proof of Theorem 8.2. Let $\mathrm{FK}_H(a) = \lambda = \lambda_{1,\mathrm{Dir}}(W)$ be the minimizing eigenvalue with $|W| = a$, and let f be the corresponding eigenfunction. Then $\pi_* f$ satisfies $A_G(\pi_* f) \geq \lambda \pi_* f$ and $\pi_* f$ is nonnegative and supported on $\pi(W)$, so

$$\lambda_{1,\mathrm{Dir}}\big(\pi(W)\big) \geq \frac{(A_G \pi_* f, \pi_* f)}{(\pi_* f, \pi_* f)} \geq \frac{(\lambda \pi_* f, \pi_* f)}{(\pi_* f, \pi_* f)} \geq \lambda.$$

Furthermore, $\pi(W)$ is a set of size at most a. Hence

$$\mathrm{FK}_G(a) \geq \lambda = \mathrm{FK}_H(a). \qquad □$$

Putting Proposition 8.1 and Theorem 8.2 together we obtain the following theorem.

THEOREM 8.4. *Let G be a d-regular graph, and let H be a cover of G. Then*

$$\lambda_2(A_G) \geq \mathrm{FK}_H(a) - \frac{da}{|V_G|}.$$

For an application to coding theory we observe the following proposition.

PROPOSITION 8.5. *Let H be the n-dimensional hypercube. Then for $\alpha \in (0,1)$ fixed we have* $\mathrm{FK}_H(2^{\alpha n}) \geq 2\sqrt{\gamma(1-\gamma)}n + o(n)$, *where* $\alpha = H_2(\gamma)$.

Proof. We take a ball of size roughly $2^{\alpha n}$. For the details see the appendices. □

Notice that we could also give a simple bound of $\mathrm{FK}_H(2^{\alpha n}) \geq \alpha n$ by taking the characteristic function of a subcube of dimension αn.

A corollary is the first MRRW bound.

COROLLARY 8.6. *For any $\delta \in (0,1)$ we have*

$$R_{\max}(\delta) \leq h\Big(1/2 - \sqrt{\delta(1-\delta)}\Big).$$

Proof. Fix an $\alpha \in (0,1)$ and a code C of information rate $\geq \alpha$ and a corresponding covering map $\pi\colon H \to G$. We apply Theorem 8.4 with $a = 2^{\alpha n}/\log n$. We conclude

$$\lambda_2(A_G) \geq 2\sqrt{\gamma(1-\gamma)}\; + o_n(1),$$

where $\alpha = h(\gamma)$. Hence

$$\delta \leq 1/2 - \sqrt{\gamma(1-\gamma)},$$

and so

$$\gamma \leq 1/2 - \sqrt{\delta(1-\delta)}$$

and the corollary follows. □

Remark 8.7. Notice that a (sub)cube of dimension αn has largest adjacency eigenvalue αn. This implies that $\mathrm{FK}_H(2^{\alpha n}) \geq \alpha n$. This gives the weak corollary that $\alpha_{\max}(\delta) \leq (1-\delta)/2$, which agrees asymptotically with the Plotkin and Griesmer bounds of coding theory (see [vL99]).

Remark 8.8. The approach which was used in this section borrows some ideas from [Nil91, Fri93]. Assume that G has a cover graph H. If f_H is a nonnegative "approximate eigenfunction" on H, we can try to form "versions" of it, f_G, on a quotient, G, with similar properties. In this section we have formed our version on G by "summing over fibers" (this was the push forward function defined above); this is a similar technique to that used by Nilli (see [Nil91]), later refined by Friedman and Kahale (see [Fri93]).[4] Our improvement on this technique was obtained by "projecting out the constants," meaning that we project out the $\vec{1}$ component from f_G rather than setting up f_G (or f_H) with a matching nonpositive component to make it orthogonal to $\vec{1}$ (as done by Nilli, Friedman, and Kahale). We explain how this improves the classical Alon–Boppana bound in the next section.

9. Classical Alon–Boppana. In this section we comment on how to improve the classical Alon–Boppana theorem by a constant factor in the second-order term. The bounds as derived in [Nil91], [Fri93], and [Kah93] all come up with a function, f, orthogonal to the constant function, whereupon we estimate $\mathcal{R}(f)$ and use $\lambda_2 \geq \mathcal{R}(f)$. In all cases, and in this paper, the following construction is involved: for a vertex, u, and distance r, we consider a function $g = g_{u,r}$ that (1) vanishes on vertices of distance greater than r to u, and (2) is constructed either as a radial function (a function of the distance) with respect to u, or is the push forward of a radial function of a lift of u on the universal cover, the infinite, d-regular tree (see also the comments at the end of section 8); therefore, g is nonnegative. (In a sense, the "optimal" g to take is the Dirichlet eigenfunction on the tree, as in [Fri93].) There are two approaches to constructing f from these g's.

The first approach is to take two vertices, u, v, of large distance apart, to take r with $g_{u,r}, g_{v,r}$ of disjoint support, and to form f as a difference of appropriate positive multiples of $g_{u,r}$ and $g_{v,r}$. Here r is roughly $(1/2)\log_{d-1} n$. This approach is taken in [Nil91] and [Fri93].

The second approach is to take one vertex, u, and to form f by taking $g_{u,r'}$ and project out the constants. In this case r' can be taken to be as large as $\log_{d-1}\bigl(n/\omega(n)\bigr)$, where $\omega(n)$ is any function with $\omega(n)/\log^2 n \to 0$ as $n \to \infty$. So r' is roughly twice as large as r, which gives an improvement by a factor of 4 in the second-order term. For example, in [Fri93] we have a bound of

$$\lambda_2 \geq 2\sqrt{d-1}\Bigl(1 - c\log_{d-1}^{-2} n + O\bigl(\log\log n \log^{-3} n\bigr)\Bigr), \tag{9.1}$$

where $c = 2\pi^2$. The projecting technique improves this to $c = \pi^2/2$. This projecting technique was used by Kahale in [Kah93]; he did not estimate c explicitly, although his choice of parameters in the proof gives $c = 2\pi^2$ (as in [Fri93]); however, it is easy to modify Kahale's choice of parameters (i.e., choose l to be the floor of $\log_{d-1}(n/\log^3 n)$ in his proof of Corollary 1 in section 3) to obtain $c = \pi^2/2$.

[4] Actually, the previous technique (of Nilli, Friedman, and Kahale) takes radial functions on G given by the radial function on H that gives the first Dirichlet eigenfunction of a ball in H of a given radius. The technique used here "pushed down" the eigenfunction on H to G by summing over the fibers, i.e., for each vertex, $v \in V_G$, we sum the eigenfunction over $\pi^{-1}(v)$. This may be better suited in certain situations, e.g., when the graphs are not regular.

We now state the improved classical Alon–Boppana bound in a theorem.

THEOREM 9.1. *Let d be fixed. For any d-regular graph on n vertices we have that* (9.1) *holds with* $c = \pi^2/2$.

Proof. We give two proofs, omitting some minor calculations. The first is to modify Kahale's proof as described above. The second proof is to apply Theorem 8.4 with H the d-regular tree and a being the floor of $n/\log^3 n$. We remark that $\mathrm{FK}_H(a)$ is at least that of the first Dirichlet eigenvalue of a ball in H of size at most a. Proposition 3.2 of [Fri93] computes this eigenvalue exactly. This changes the value of k in Proposition 3.7 of [Fri93] (which is the radius of the ball to which Proposition 3.2 of [Fri93] is applied) from $(1/2)\log_{d-1} n$ to essentially double that, i.e., the floor of $\log_{d-1}(n/\log^3 n)$. Thus the second-order term of $2\sqrt{d-1}\pi^2/(2k^2)$ essentially gets multiplied by $1/4$ (up to higher-order terms). The only additional term here is the $da/|V_G|$ term in Theorem 8.4; this term is $O(\log^{-3} n)$ by our choice of a. □

10. A stronger Alon–Boppana bound.

10.1. A simple improvement.

In this subsection we give an example of a more general generalized Alon–Boppana bound. Namely, the following theorem and corollary strengthen Proposition 8.1.

THEOREM 10.1. *Let G be a d-regular graph and let p be any real-valued function defined on the eigenvalues of A_G. Then*

$$(f,f)\max_{i\ge 2} p(\lambda_i) \ge \bigl(p(A_G)f,f\bigr) - p(d)(f,\vec{1})^2/|V_G|.$$

The theorem follows immediately from the spectral decomposition on A_G as applied to f.

COROLLARY 10.2. *If in addition to the hypothesis in the above theorem we have that f is supported in U, then*

$$(f,f)\max_{i\ge 2} p(\lambda_i) \ge \bigl(p(A_G)f,f\bigr) - p(d)(f,f)|U|/|V_G|.$$

The special case $p(x) = x$ was the bound used in the previous section. When G has a cover which is distance regular, then there is a very natural choice of polynomials in the corollary which enables us to have some control on the term $\bigl(p(A_G)f,f\bigr)$ when $f = \chi_v$ for any vertex v of G. Indeed, let H be a distance regular cover of G. Let D denote the diameter of H. Then there are $D+1$ polynomials $P_0, P_1, P_2, \dots, P_D$ (see [BCN89]) such that $P_i(A_H)$ is the adjacency matrix of the graph with the same vertices as H and two vertices are joined by an edge iff they are at distance i in H. In such a setting for any $Q = \sum_{i=0}^{D} \beta_i P_i$, where the β_i's are nonnegative we have that $\bigl(Q(A_G)\chi_v,\chi_v\bigr) \ge \bigl(Q(A_H)\chi_v,\chi_v\bigr)$. Notice now that $\bigl(Q(A_H)\chi_v,\chi_v\bigr) = \beta_0$ and, therefore,

$$\bigl(Q(A_G)\chi_v,\chi_v\bigr) \ge \beta_0. \tag{10.1}$$

The coset graphs associated to a binary linear code of length n that we consider in this article have a common cover which is distance regular, namely, the boolean cube $\mathbb{B}^n$. An application of the aforementioned remark leads to the Delsarte linear programming bound in coding theory as explained in the following subsection.

10.2. Connections with the Delsarte approach. Let us first quickly review the linear programming approach for obtaining upper bounds on the minimum distance of a code (see [MS77v1, MS77v2, vL99] for more details). For a code $C \in (\mathbb{F}_2)^n$, we consider the *distance distribution* of the code, i.e., the B_i's for $i = 0, \dots, n$, where B_i denotes the average number of codewords of distance i to a fixed codeword, that is, $B_i \overset{\text{def}}{=} \frac{1}{|C|} \left|\{(x,y); x \in C, y \in C, d(x,y) = i\}\right|$. The linear programming bound is based on the inequality

$$\sum_{i=0}^{n} B_i K_k(i) \geq 0$$

for $k \in \{0, 1, \dots, n\}$, where K_k is a Krawtchouk polynomial of degree k:

$$K_k(x) \overset{\text{def}}{=} \sum_{j=0}^{k} (-1)^j \binom{x}{j} \binom{n-x}{k-j}$$

with $\binom{x}{j} \overset{\text{def}}{=} \frac{x(x-1)\dots(x-j+1)}{j!}$. This yields linear inequalities which should be satisfied by the B_i's. By maximizing the sum of the B_i's (which is equal to the size of the code) which satisfy these inequalities we obtain a linear programming problem for which an upper bound can be found by duality. This duality result can be written as the following (see Theorem 5.3.5 of [vL99]).

THEOREM 10.3. *Let $\beta(x) = 1 + \sum_{k=1}^{n} \beta_k K_k(x)$ be any polynomial with $\beta_k \geq 0$ ($1 \leq k \leq n$) such that $\beta(j) \leq 0$ for $j = d, d+1, \dots, n$; then any code of minimum distance $\geq d$ and length n has cardinality at most $\beta(0)$.*

Finding interesting choices for β turns out to be a nontrivial task; however, the first MRRW bound can be obtained by a direct application of this theorem by choosing β appropriately (see [vL99]).

We now claim that this theorem is a simple consequence of Corollary 10.2, provided we restrict to linear codes. Indeed, if we let $P_k \overset{\text{def}}{=} K_k((n-x)/2)$, then $P_k(A_{\mathbb{B}^n})$ is nothing but the adjacency matrix of the graph with vertices belonging to ${\mathbb{F}_2}^n$ and two vertices being adjacent iff they are at Hamming distance k. This follows immediately from classical results about the Hamming association scheme (see, for instance, Chapter 21 in [MS77v1, MS77v2]). Therefore, by using the remark which follows Corollary 10.2 for any polynomial $Q(x) = 1 + \sum_{k=1}^{n} \beta_k P_k(x) = 1 + \sum_{k=1}^{n} \beta_k K_k((n-x)/2)$ with $\beta_k \geq 0$ we have that for any vertex of the coset graph G of a binary linear code of length n,

$$\big(Q(A_G)\chi_v, \chi_v\big) \geq 1. \tag{10.2}$$

Notice now that by Theorem 5.1 $P_k(\lambda_i) = K_k(j)$ for some integer $j \in [d_{\min}(C), n]$ for any eigenvalue of the adjacency matrix of G different from n. Therefore, $Q(\lambda_i) = 1 + \sum_{k=1}^{n} \beta_k K_k(j)$. This implies that

$$(\chi_v, \chi_v) \max\{Q(\lambda_i) | 2 \leq i \leq |V_G|\} \leq 0 \tag{10.3}$$

if Q has been chosen such that $1 + \sum_{k=1}^{n} \beta_k K_k(j) \leq 0$ for any integer $j \in [d_{\min}(C), n]$ (since this implies that $Q(\lambda_i) \leq 0$ for $i \in \{2, \dots, |V_G|\}$). We eventually obtain by using Corollary 10.2 with $f = \chi_v$ and by putting inequalities (10.3) and (10.2) together that

$$0 \geq 1 - \frac{1 + \sum_{k=1}^{n} \beta_k P_k(0)}{|C|}$$

since $Q(n) = 1 + \sum_{k=1}^{n} \beta_k P_k(0)$ by Theorem 5.1 and $|V_G| = |C|$. This proves Theorem 10.3.

11. General graph theory.

In this section we review some basic terminology and notions needed to generalize covering theory to graphs with multiple edges and/or self-loops.

11.1. Directed graphs.

By a *directed graph* we mean a pair of sets, $G = (V_G, E_G)$, with an identification of E_G as a multiset of $V_G \times V_G$. In other words, G comes with an *incidence map* $i_G \colon E_G \to V_G \times V_G$. We write i, E, V for i_G, E_G, V_G if no confusion can result. If $i(e) = (u, v)$, we say that e is of *type* (u, v) or that e *originates* in u and *terminates* in v or that e's *tail* is u and e's *head* is v; in any case we will write $e \sim (u, v)$; if no multiple edges occur, i.e., if i is injective, then we may unambiguously write $e = (u, v)$.

A *walk* is an alternating sequence of vertices and edges such that when $\ldots, v, e, \ldots$ occurs in the sequence, e's tail is v, and similarly with the order of v, e reversed (with "head" replacing "tail"). The *adjacency matrix*, A_G, of a graph, G, is the square matrix indexed on V_G whose u, vth entry counts how many edges have type u, v. For a positive integer, k, the u, vth entry of $(A_G)^k$ counts how many directed walks there are from u to v of length k. All this makes sense if V_G or E_G is infinite, although the entries of A_G or $(A_G)^k$ may not be finite.

A morphism $\pi \colon H \to G$ of directed graphs is a collection of maps $\pi_V \colon V_H \to V_G$ and $\pi_E \colon E_H \to E_G$ that commutes with the incidence relations (i.e., $i_G \circ \pi_E = (\pi_V \times \pi_V) \circ i_H$). We often drop the subscripts from π_V, π_E if no confusion can result.

For a morphism of directed graphs, $\pi \colon H \to G$, it is possible to give a number of equivalent definitions for π to be a *covering map*; all definitions amount to π being a local isomorphism in some sense. One definition is that for every vertex, $v \in V_G$, $w \in \pi^{-1}(v)$, and every edge $e \in E_G$ with tail v, there is exactly one $f \in \pi^{-1}(e)$ whose tail is w, and similarly with "head" replacing "tail." Another possibility is to define the *geometric realization* of a graph (as in [Fri93]); then a covering map is a covering map in the topological sense.

11.2. Graphs.

By an *undirected graph* or simply a *graph* we mean a directed graph, G, with an involution[5] ι on E_G that reverses heads and tails; in other words, G's edges are paired, $e \sim (u, v)$ with an edge $\iota(e) \sim (v, u)$, where e may be paired with itself[6] if $u = v$.

A morphism of graphs is one of the underlying directed graphs that commutes with the ι's. Covering maps, adjacency matrices, and walks in graphs are just the same as that of the underlying directed graphs.

It is now simple to see that all the theorems of this paper could as well have been stated for graphs that may have self-loops or multiple edges.

12. Concluding remarks.

One of the most exciting problems to us is that of finding the Faber–Krahn maximizer and maximum of the hypercube. One can find examples[7] of very small or large balls that are *not* the Faber–Krahn minimizers.

[5] For ι to be an involution means that $\iota \circ \iota$ is the identity.

[6] This gives rise to "half-loops," which are edges paired with themselves, and "whole-loops" in the language of [Fri93]. For example, a whole-loop contributes 2 to an entry on the diagonal of the adjacency matrix, whereas a half-loop contributes 1.

[7] For example, in the 3-hypercube, the two-dimensional subcube has eigenvalue 2, which is greater than that of a ball of the same size, namely, $\sqrt{3}$. Similarly for the three-dimensional ball in the 7-hypercube. Also the $n/2 - \sqrt{n}$ radius ball has eigenvalue $n - 4$ (since $n/2 - \sqrt{n}$ is the first zero of the

Question 12.1. Given $\gamma \in (0, 1/2)$, is

$$\varliminf_{n\to\infty} \mathrm{FK}_{\mathbb{B}^n}\bigl(2^{H_2(\gamma)}\bigr)/n = 2\sqrt{\gamma(1-\gamma)},$$

i.e., are balls asymptotically maximizers for the hypercube? If the answer is no, then according to the method of Corollary 8.6, we have an improvement to the first MRRW bound.

Appendix A. Calculations for coding theory. In this section we derive some simple combinatorial bounds needed in our discussion of coding theory bounds.

Throughout this section we write $f(n) \approx g(n)$ if $\bigl(\log f(n)\bigr)/\bigl(\log g(n)\bigr) \to 1$ as $n \to \infty$ (for example, $n \approx 2n$ but $n \not\approx n^2$).

Lemma A.1. *If $\rho \in (0, 1/2]$ is fixed, and if any integer $n > 0$, we set $r = r(n) = \lfloor \rho n \rfloor$; then*

$$|B_r| \approx \binom{n}{r} \approx 2^{nh(\rho)},$$

where $|B_r|$ is the size of the ball of radius r in the n-hypercube, and where

$$h(\theta) = -\theta \log_2 \theta - (1-\theta)\log_2(1-\theta).$$

Proof. This is a very standard application of Stirling's formula; see, for example, [vL99]. □

Lemma A.2. *Let $\alpha \in (0, 1)$ be fixed. For any integer $n > 0$ set k to be the even integer equal either to $\lfloor \alpha n \rfloor$ or to $\lfloor \alpha n \rfloor + 1$. Then*

$$N_k(\mathbb{B}^n) \approx 2^{h(\beta_0)n-n} n^k (1-2\beta_0)^k,$$

where β_0 is the unique solution in $(0, 1/2)$ to the following equation:

$$(1-2\beta_0)\log(\beta_0^{-1}-1) = 2\alpha. \tag{A.1}$$

Proof. Since $A_{\mathbb{B}^n}$ has eigenvalues $n - 2i$ with multiplicity $\binom{n}{i}$, we have

$$N_k(\mathbb{B}^n) = \frac{1}{2^n}\sum_{i=0}^{n}\binom{n}{i}|n-2i|^k.$$

It follows that setting $B_i = \binom{n}{i}(n-2i)^k$, we see that

$$\frac{N_k(\mathbb{B}^n)}{n+1} \le \max_{i=0,\dots,n/2} \frac{B_i}{2^n} \le N_k(\mathbb{B}^n).$$

To find the i maximizing B_i, we write

$$\frac{B_{i+1}}{B_i} = \left(\frac{n-i}{i+1}\right)\left(\frac{n-2(i+1)}{n-2i}\right)^k = \left(\frac{n-i}{i+1}\right)e^{k\log_e(1-2/(n-2i))}.$$

Set $\beta = \beta_n = i_0/n$, where i_0 is the (an) $i \le n/2$ maximizing B_i. Since $B_{i_0+1} < B_{i_0}$ we have

$$\left(\frac{1-\beta}{\beta} + O(n^{-1})\right)e^{\frac{-2\alpha}{1-2\beta}+O(n^{-1})} < 1.$$

second Krawtchouk polynomial), and the $(n-4)$-dimensional subcube is smaller; so by monotonicity (see [Fri93]) the ball here can also be beaten.

Hence

$$\frac{1-\beta}{\beta} < e^{\frac{2\alpha}{1-2\beta}} + O(n^{-1}).$$

Similarly $B_{i_0-1} < B_{i_0}$, and the reverse inequality holds. Taking logarithms, we conclude that

$$\log(\beta^{-1}-1)(1-2\beta) = 2\alpha,$$

where β is the lim sup and lim inf of β_n. But differentiation shows that

$$f(\beta) = \log(\beta^{-1}-1)(1-2\beta)$$

has $f'(\beta) = -2\log(\beta^{-1}-1) - (1-2\beta)/(\beta-\beta^2)$ which is less than 0 for $\beta \in (0,1/2)$. It follows that there is a unique $\beta_0 \in (0,1/2)$ that satisfies (A.1), and this β_0 is the limit of β_n. □

COROLLARY A.3. *For α, n, k as above we have*

$$N_k(\mathbb{B}^n) \approx n^k \left(\frac{\alpha+\omega(\alpha)}{e}\right)^{k/2},$$

where $\omega(\alpha)$ is a function of α with $\omega(\alpha) = O(\alpha^{3/2})$ as $\alpha \to 0$.

Proof. For $\beta = 1/2 - \epsilon$ with ϵ small we have

$$\log(\beta^{-1}-1)(1-2\beta) = \log\left(\frac{1/2+\epsilon}{1/2-\epsilon}\right)2\epsilon = 2\epsilon\log\bigl(1+4\epsilon+O(\epsilon^2)\bigr) = 8\epsilon^2 + O(\epsilon^3).$$

Hence for α small we have

$$2\alpha = 8\epsilon^2 + O(\epsilon^3) \quad \text{or} \quad \sqrt{\alpha/4} = \epsilon + O(\epsilon^2) = \epsilon + O(\alpha).$$

Differentiation shows that

$$h'(x) = \log_2(x^{-1}-1), \quad h''(x) = \frac{-\log_2 e}{x-x^2}.$$

So $h'(1/2) = 0$ and $h''(1/2) = -4\log_2 e$, and

$$h(1/2-\epsilon) = 1 - 2(\log_2 e)\epsilon^2 + O(\epsilon^3).$$

It follows that

$$2^{-n}2^{nh(\beta)}(n-2\beta)^k \approx 2^{-n(2\log_2 e)\epsilon^2 + O(n\epsilon^3)} n^k (1-2\epsilon)^k$$

$$\approx e^{-2n(\alpha/4+O(\alpha^{3/2}))} n^{\alpha n} \bigl(2\sqrt{\alpha/4} + O(\alpha)\bigr)^{\alpha n}$$

$$\approx e^{-\alpha n/2} e^{O(\alpha^{3/2}n} \bigl(n\sqrt{\alpha}\bigr)^{\alpha n} \bigl(1+\bigl(\sqrt{\alpha}\bigr)\bigr)^{\alpha n} \approx n^k(\alpha/e)^{k/2}\bigl(1+O\bigl(\sqrt{\alpha}\bigr)\bigr)^{k/2},$$

and the proposition is finished. □

Proof of Corollary 6.5. Let G be the coset graph of $C^{\perp}$. Consider the largest odd integer, k, for which

$$\sum_{i=0}^{k+2} \binom{n}{i} \geq 2^{\alpha n}. \tag{A.2}$$

It follows that there are two points in G of distance $\geq k+3$, and hence two edges of distance $\geq k+1$. By Theorem 6.3, we have

$$\lambda_2(A_G) \geq \bigl(N_{k-1}(\mathbb{B}^n)\bigr)^{1/k}.$$

But by (A.2) and Lemma A.1, we have

$$2^{nh(k/n)+O(1)} \approx 2^{\alpha n},$$

and thus

$$k/n = h^{-1}(\alpha) + o_n(1)$$

(where $o_n(1)$ denotes a function that tends to zero as $n \to \infty$). Since $h^{-1}(\alpha) = \alpha/\log_2(1/\alpha) + O(\alpha)$ for α small, Corollary A.3 then implies that

$$\lambda_2/n \geq \sqrt{\frac{\alpha}{e\log_2(1/\alpha)}} + \omega(\alpha) + o_n(1),$$

where $\omega(\alpha) = O\bigl(\sqrt{\alpha}\bigr)$. Now we use the fact that the minimum distance is $(n-\lambda_2)/2$. □

Proof of Corollary 7.3. Let k be as in the previous proof, except that k is the largest odd integer such that

$$N_{k+1}(\mathbb{B}^n) \geq n^k/|V_G|.$$

Then taking kth roots and dividing by n yields that $k = n\gamma + o(n)$, where

$$\sqrt{\gamma/e} + \omega(\gamma) = 2^{-\alpha/\gamma},$$

where $\omega(\gamma) = O(\gamma)$ for γ small. Hence

$$\gamma = \frac{2\alpha}{\log_2(1/\alpha)} + O(\alpha)$$

for α small. Now we follow as in the proceeding proof, except that here $\gamma = k/n$ is, to first order, twice what it was in the previous proof; this factor of 2 changes the C from $1/\sqrt{4e}$ to $1/\sqrt{2e}$ here. □

Appendix B. A calculus proposition. In this section we prove Proposition 4.1.

Let f be defined at α_0, and set $\delta_0 = f(\alpha_0)$. It suffices to show that $\alpha_{\max}(\delta_0) \leq \alpha_0$.

For any $\epsilon > 0$ near 0, fix an $\eta > 0$. If

$$\alpha_{\max}\bigl(f(\alpha_0+\epsilon)+\eta\bigr) > \alpha_0 + \epsilon, \tag{B.1}$$

then there are codes C_i of length $n_i \to \infty$ as $i \to \infty$ such that $\delta_{C_i} \geq f(\alpha_0 + \epsilon) + \eta$ and

$$\overline{\lim_{i\to\infty}} \alpha_{C_i} > \alpha_0 + \epsilon.$$

By passing to a subsequence we can assume that $\alpha_{C_i} > \alpha_0 + \epsilon$ for all i. But then $\delta_{\max}$ exceeds f (by at least η) at the value $\alpha_0 + \epsilon$, which is impossible. So inequality (B.1) is impossible, meaning that

$$\alpha_{\max}\bigl(f(\alpha_0 + \epsilon) + \eta\bigr) \leq \alpha_0 + \epsilon.$$

Now let $\eta = \eta(\epsilon) = \delta_0 - f(\alpha_0 + \epsilon)$ and let $\epsilon \to 0$. We conclude $\alpha_{\max}(\delta_0) \leq \alpha_0$, and that completes the proof.

Appendix C. The first eigenvalue of a ball and related calculations. In this appendix we prove Proposition 8.5.

Since the size of the ball of radius $n\gamma$ in $\mathbb{B}^n$ is $\approx 2^{nh(\gamma)}$, we need only show that the ball of radius $n\gamma$ has first eigenvalue at least

$$2n\sqrt{\gamma(1-\gamma)}\ + o(n).$$

Let 0 denote the origin in $(\mathbb{F}_2)^n$, which is a vertex of $\mathbb{B}^n$; the *weight*, $|v|$, of a vertex, v, of $\mathbb{B}^n$ is its distance from 0, or the number of nonzero coordinates it has. Consider those functions, f, on $\mathbb{B}^n$ that depend only on the weight of the vertex. For such an f, let f_{nrm}, the *normalization of* f, be the function on $[0, \ldots, n]$ such that

$$f_{\mathrm{nrm}}(i) = f(v) \Bigg/ \sqrt{\binom{n}{i}} \qquad \text{for any } v \text{ with } |v| = i.$$

Then it is easy (and completely standard) to see that

$$(A_G f)_{\mathrm{nrm}}(i) = \sqrt{i(n-i+1)}\, f_{\mathrm{nrm}}(i-1) + \sqrt{(i+1)(n-i)}\, f_{\mathrm{nrm}}(i+1)$$

for all i (the coefficient of the right-hand side vanishes for f_{nrm} at the values -1 and $n+1$). So under normalization the operator A_G becomes a symmetric tridiagonal operator $\widetilde{A}$ whose $i-1, i$ entry is $\sqrt{i(n-i+1)}$. It follows that if $i \in [\gamma n - \omega(n), \gamma n]$, where $\omega(n)$ is any function that is $o(n)$, then the $i-1, i$ entry is

$$n\sqrt{\gamma(1-\gamma)}\ + o(n).$$

Hence, by monotonicity (see, e.g., [Fri93]), the first Dirichlet eigenvalue of the ball of radius $n\gamma$ is at least that of the path of length $\omega(n) + 1$ times

$$n\sqrt{\gamma(1-\gamma)}\ + o(n).$$

But this path's eigenvalue is well known to be $2\cos(\pi/\omega(n))$, giving us a lower bound on the ball's eigenvalue of

$$2n\sqrt{\gamma(1-\gamma)}\ + o(n),$$

provided that $\omega(n)$ grows faster than $\sqrt{n}$ (e.g., we may take $\omega(n) = n^{3/4}$). □

REFERENCES

[BCN89] A. E. Brouwer, A. M. Cohen, and A. Neumaier, *Distance Regular Graphs*, Springer-Verlag, Berlin, 1989.

[Cha84] I. Chavel, *Eigenvalues in Riemannian Geometry*, Academic Press, Orlando, FL, 1984 (including a chapter by Burton Randol with an appendix by Jozef Dodziuk).

[DS91] C. Delorme and P. Solé, *Diameter, covering index, covering radius and eigenvalues*, European J. Combin., 12 (1991), pp. 95–108.

[Fab23] C. Faber, *Beweis, dass unter allen homogenen Membrane von gleicher Spannung die Kreisförmige den tiefsten Grundton gibt*, Sitzungsber—Bayer. Akad. Wiss., Math.-Phys. Munich, 1923, pp. 169–172.

[Fri93] J. Friedman, *Some geometric aspects of graphs and their eigenfunctions*, Duke Math. J., 69 (1993), pp. 487–525.

[Fri03] J. Friedman, *Relative expanders or weakly relatively Ramanujan graphs*, Duke Math. J., 118 (2003), pp. 19–35.

[Kah93] N. Kahale, *On the second eigenvalue and linear expansion of regular graphs*, in Expanding Graphs (Princeton, NJ, 1992), DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 10, AMS, Providence, RI, 1993, pp. 49–62. A preliminary version appears in Proceedings of the 33rd Annual Symposium on Foundations of Computer Science, IEEE, 1992, pp. 296–303.

[Kra25] E. Krahn, *Über eine von Rayleigh formulierte Minimaleigenschaft des Kreises*, Math. Annales, 94 (1925), pp. 97–100.

[LPS88] A. Lubotzky, R. Phillips, and P. Sarnak, *Ramanujan graphs*, Combinatorica, 8 (1988), pp. 261–277.

[MRRW77] R. J. McEliece, E. R. Rodemich, H. Rumsey, Jr., and L. R. Welch, *New upper bounds on the rate of a code via the Delsarte–MacWilliams inequalities*, IEEE Trans. Inform. Theory, IT-23 (1977), pp. 157–166.

[MS77v1] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, I, North-Holland Math. Library 16, North-Holland, Amsterdam, 1977.

[MS77v2] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, II, North-Holland Math. Library 16, North-Holland, Amsterdam, 1977.

[Nil91] A. Nilli, *On the second eigenvalue of a graph*, Discrete Math., 91 (1991), pp. 207–210.

[vL99] J. H. van Lint, *Introduction to Coding Theory*, 3rd ed., Springer-Verlag, Berlin, 1999.

SIAM J. DISCRETE MATH.
Vol. 19, No. 3, pp. 719–727

THE STRONG CHROMATIC INDEX OF RANDOM GRAPHS*

ALAN FRIEZE[†], MICHAEL KRIVELEVICH[‡], AND BENNY SUDAKOV[§]

Abstract. The strong chromatic index of a graph G, denoted by $\chi_s(G)$, is the minimum number of colors needed to color its edges so that each color class is an induced matching. In this paper we analyze the asymptotic behavior of this parameter in a random graph $G(n,p)$, for two regions of the edge probability $p = p(n)$. For the dense case, where p is a constant, $0 < p < 1$, we prove that with high probability $\chi_s(G) \leq (1 + o(1))\frac{3}{4}\frac{n^2 p}{\log_b n}$, where $b = 1/(1-p)$. This improves upon a result of Czygrinow and Nagle [*Discrete Math.*, 281 (2004), pp. 129–136]. For the sparse case, where $np < \frac{1}{100}\sqrt{\log n/\log\log n}$, we show that with high probability $\chi_s(G) = \Delta_1(G)$, where $\Delta_1(G) = \max\{d(u) + d(v) - 1 : (u,v) \in E(G)\}$. This improves a result of Palka [*Australas. J. Combin.*, 18 (1998), pp. 219–226].

Key words. strong chromatic index, random graphs

AMS subject classifications. 05C80, 05C15

DOI. 10.1137/S0895480104445757

1. Introduction.

Given a graph $G = (V,E)$, the *strong chromatic index* $\chi_s(G)$ is the minimum number of colors needed to color the edges of G so that every color class is an *induced* matching; i.e., any two edges of the same color are at distance at least 2 in G. This notion was introduced by Erdős and Nešetřil (see [3]). Equivalently, it is the chromatic number of the square $L(G)^2$ of the line graph $L(G)$. Thus if Δ denotes the maximum degree of G, the maximum degree of $L(G)^2$ is at most $2\Delta^2 - 2\Delta$ and so $\chi_s(G) \leq 2\Delta^2 - 2\Delta + 1$. It was conjectured in [3] that $\chi_s(G) \leq 5\Delta^2/4$ and this would be tight if true. Using a probabilistic argument, Molloy and Reed [5] showed that $\chi_s(G) \leq (2 - \varepsilon)\Delta^2$ for some small positive constant ε.

In this paper we study the strong chromatic index of the random graph $G(n,p)$. As usual, $G(n,p)$ stands for the probability space of labeled graphs on n vertices, where every edge appears independently and with probability $p = p(n)$. Palka [6] showed that if $p = \Theta(n^{-1})$, then whp[1] $\chi_s(G) = O(\Delta(G)) = O(\log n/\log\log n)$. Vu [7] showed that if $n^{-1}(\log n)^{1+\delta} \leq p \leq n^{-\varepsilon}$ for constants $0 < \varepsilon, \delta < 1$, then whp $\chi_s(G) = O(\Delta^2/\log\Delta)$. Czygrinow and Nagle [2] showed that if $p > n^{-\varepsilon}$, then $\chi_s(G) \leq (1 + o(1))n^2 p/\log_b n$, where $b = 1/(1-p)$. In this paper we will obtain new bounds on $\chi_s(G(n,p))$ that improve the above results of Palka and of Czygrinow and Nagle.

To formulate our first theorem we need the following definition. For graph $G =$

*Received by the editors August 23, 2004; accepted for publication (in revised form) April 26, 2005; published electronically November 23, 2005.
http://www.siam.org/journals/sidma/19-3/44575.html

[†] Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213 (alan@random.math.cmu.edu). This author's research was supported in part by NSF grant CCR-0200945.

[‡] Department of Mathematics, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel (krivelev@post.tau.ac.il). This author's research was supported in part by USA–Israel BSF grant 2002-133 and by grant 64/01 from the Israel Science Foundation.

[§] Department of Mathematics, Princeton University, Princeton, NJ 08544 (bsudakov@math.princeton.edu). This author's research was supported in part by NSF grants DMS-0355497 and DMS-0106589 and by an Alfred P. Sloan fellowship.

[1] A sequence of events $\mathcal{E}_n$ occurs *with high probability* (whp) if $\lim_{n\to\infty} \mathbf{Pr}(\mathcal{E}_n) = 1$.

(V, E) let $d(v)$ denote the degree of vertex $v \in V$ and let

$$\Delta_1 = \Delta_1(G) = \max\{d(u) + d(v) - 1 : \ (u, v) \in E\}.$$

Set

$$\lambda = \left(\frac{\log n}{\log\log n}\right)^{1/2}.$$

Then, for the sparse random graphs we prove the following tight result.

THEOREM 1. *Let p be such that $np \le \lambda/100$. Then whp, with $G = G(n, p)$,*

$$\chi_s(G) = \Delta_1(G).$$

Remark 1. A straightforward calculation shows that in this range of edge probabilities, $\Delta_1(G) = (1 + o(1))\Delta(G)$.

Remark 2. The observant reader will notice that our proof shows that the related *choice* number is also Δ_1 whp; i.e., as long as each edge is given a list of Δ_1 colors, we can strongly edge color it.

We have learned via private communication with Tomasz Łuczak that in unpublished work he has obtained a result similar to Theorem 1.

For the dense case we improve the aforementioned result of Czygrinow and Nagle by a constant factor.

THEOREM 2. *Let $p > 0$ be a constant. Denote $b = 1/(1 - p)$. Then whp, with $G = G(n, p)$,*

$$\chi_s(G) \le (1 + o(1))\frac{3}{4}\frac{n^2 p}{\log_b n}.$$

By the above result, the edges of $G(n, p)$ can be a.s. strongly colored so that the average size of a color class is at least $(1 - o(1))\frac{2}{3}\log_b n$.

Remark 3. The size of the largest induced matching in $G(n, p)$ is whp asymptotically equal to $\log_b n$, and so whp $\chi_s(G)$ is asymptotically at least $\frac{n^2 p}{2\log_b n}$.

1.1. Notation. A sequence of events $\mathcal{E}_n$ is said to occur *quite surely* (qs) if $\mathbf{Pr}(\mathcal{E}_n) = O(n^{-K})$ for any constant $K > 0$.

Unless the base is specifically mentioned, log will refer to natural logarithms.

We often refer to the Chernoff bound for the tails of the binomial distribution. By this we mean one of the following (see, e.g., [4]):

$$\begin{aligned} \mathbf{Pr}(B(n, p) \le (1 - \varepsilon)np) &\le e^{-\varepsilon^2 np/2}, \\ \mathbf{Pr}(B(n, p) \ge (1 + \varepsilon)np) &\le e^{-\varepsilon^2 np/3}, \qquad \varepsilon \le 1, \\ \mathbf{Pr}(B(n, p) \ge \mu np) &\le (e/\mu)^{\mu np}. \end{aligned}$$

2. Sparse random graphs. Given a graph G with maximum degree Δ, let $\beta = \sqrt{\Delta}/2$. Denote by L_β the set of vertices of G which are within distance at most 2 from the set of vertices of degree at least β. Let G_β be the subgraph of G induced by L_β. First we need the following simple statement.

LEMMA 3. *Let G be a graph for whose subgraph G_β is acyclic. Then $\chi_s(G) = \Delta_1(G)$.*

Proof. Clearly, for every edge (u, v) of G, the edge itself and all edges incident with u, v must have distinct colors. Therefore $\chi_s(G) \geq \Delta_1(G)$ and it remains to show the reverse inequality.

We start by coloring the edges of G_β. Since all connected components of this graph are trees, it is enough to show that edges of every such tree T can be colored using only $\Delta_1(T) \leq \Delta_1(G)$ colors. We do this by induction on the number of edges of T. It is trivial if T has one edge or, more generally, is a star. Now root T at an arbitrary vertex r and let x be a vertex of degree 1 of T at maximum distance from r. Let $y \neq r$ be its unique neighbor in T and let z be the neighbor of y on the path from x to r ($z = r$ is possible here). Let $T' = T - x$ and let d, d' refer to vertex degrees in T, T', respectively. By induction we can color the edges of T' using only $\Delta_1(T') \leq \Delta_1(T)$ colors. Then the number of colors forbidden for edge $e = (x, y)$ is at most

$$D' = d'(y) + d'(z) - 1 = \bigl(d(y) - 1\bigr) + d(z) - 1 = \bigl(d(y) + d(z) - 1\bigr) - 1 \leq \Delta_1(T) - 1.$$

Therefore there is a color which is not used at y or z and we can use it to color the edge (x, y).

Having finished coloring the edges of G_β, we can color the remaining edges $e_1, e_2, \dots, e_M$ of G in this (arbitrary) order. Note that, by definition, for every edge (u, v) outside G_β, all the neighbors of both u and v should have degree less than β. Therefore when we come to color e_i we find that at most $2\beta^2 \leq \Delta/2 < \Delta_1$ colors have been forbidden by the coloring of previous edges, and so there will always be an allowable color. □

LEMMA 4. *Let p be such that $np \leq \lambda/100$. Let T be a fixed set of vertices of size $|T| = t$ and let A be a fixed set of at most $2t$ edges. Then conditioning on the event that all edges in A are present in $G(n, p)$, the probability that all the vertices in T have degree at least $\lambda/3$ is at most $2e^{-\lambda t/10}$.*

Proof. By definition, it is easy to see that for such a set T, either there are at least $\lambda t/9$ edges in the cut $(T, V(G) - T)$, or the set T spans at least $\lambda t/9$ edges of $G(n, p)$. Since we are conditioning on the presence of at most $2t$ edges, we have that either there are at least $\lambda t/9 - 2t \geq \lambda t/10$ random edges in the cut $(T, V(G) - T)$ or, similarly, the set T contains at least $\lambda t/10$ random edges of $G(n, p)$. Using the fact that $np \leq \lambda/100$, the probability of the first event can be bounded by

$$\begin{aligned}\binom{t(n-t)}{\lambda t/10} p^{\lambda t/10} &\leq (10e(n-t)p\lambda^{-1})^{\lambda t/10} \\ &\leq e^{-\lambda t/10}.\end{aligned}$$

Similarly, the probability of the second event is at most

$$\begin{aligned}\binom{t(t-1)/2}{\lambda t/10} p^{\lambda t/10} &\leq (5e(t-1)p\lambda^{-1})^{\lambda t/10} \\ &\leq e^{-\lambda t/10}.\end{aligned}$$

Altogether we obtain that the probability that all the vertices in T have degree at least $\lambda/3$ is at most $2e^{-\lambda t/10}$. □

LEMMA 5. *Let p be such that $np \leq \lambda/100$. Then whp, with $G = G(n, p)$, the subgraph G_β is acyclic.*

Proof. If $np \leq 1/\log\log n$, then the probability that $G(n, p)$ contains a cycle is at most $\sum_{t \geq 0} n^t p^t = o(1)$; i.e., it is acyclic whp. Therefore we can assume that

$np \geq 1/\log\log n$. In this case it is well known (see, e.g., [1]) that the maximum degree of the random graph is whp at least $(1+o(1))\frac{\log n}{\log\log n}$. Let X be the set of vertices of $G = G(n,p)$ which are within distance at most 2 from a vertex of degree at least $d_0 = \lambda/3$. Then it is enough to show that the subgraph of $G(n,p)$ induced by X is acyclic whp.

Let C be a shortest cycle in the subgraph $G[X]$ induced by X, and let t be the length of C. We claim that there are at least $t/10$ vertex disjoint paths of length at most 2 connecting vertices of the cycle to vertices of degree at least d_0. Since every vertex of the cycle is within distance at most 2 from some vertex of degree at least d_0, there is always at least one such path. Therefore we can assume that $t \geq 10$.

Let $v_1, \dots, v_s$ be a largest set of vertices of C such that the distance along the cycle between any two of them is at least 5. Clearly $s = \lfloor t/5 \rfloor \geq t/10$. Note that since C was the shortest cycle in $G[X]$, the distance between every pair $v_i \neq v_j$ in this graph is also at least 5. By the definition of X, for every $v_i, 1 \leq i \leq s$, there is a path P_i of length at most 2 from v_i to a vertex of degree at least d. All vertices of this path belong to X and the paths P_i and P_j are vertex disjoint, since otherwise the distance between v_i and v_j in $G[X]$ would be at most 4. The path P_i may share edges with C. On the other hand, once the path P_i leaves the C it cannot come back, since otherwise it will create a shorter cycle. Let u_i be the last vertex of P_i which still belongs to C, P'_i be the part of P_i which is edge disjoint from C, and w_i be the endpoint of P'_i which has degree at least d_0. Denote by H the union of all paths P'_i and C. We now estimate the probability that $G(n,p)$ contains such a subgraph.

The number of ways to choose a cycle C is at most n^t and the probability that it appears in $G(n,p)$ is p^t. We can choose the set of vertices u_i in at most $\binom{t}{s} \leq 2^t$ ways. The path between u_i and w_i can have length $0, 1$, or 2, and there are at most $3^{t/5}$ different ways to choose a length for every path P'_i. The number of paths of length $0, 1, 2$ is at most $1, n, n^2$, respectively, and their existence probabilities are $1, p, p^2$. Note that after we choose the paths P'_i the vertices w_i are fixed and we expose a set A of at most $t + 2(t/5) \leq 2t$ edges of $G(n,p)$. Therefore, by Lemma 4, the probability that all the vertices w_i have degree at least d_0 is bounded by $2e^{-\lambda s/10} \leq 2e^{-\lambda t/100}$. As $np < \lambda < \sqrt{\log n}$, we can combine the above facts to conclude that the probability that a graph H appears in $G(n,p)$ is bounded by

$$\begin{aligned}
&\sum_{t\geq 3} n^t\, p^t\, 2^t\, 3^{t/5}\big(1+np+n^2p^2\big)^{t/5} e^{-\lambda t/100} \\
&< \sum_{t\geq 3} (6np)^t (2\lambda)^{t/5} e^{-\lambda t/100} \\
&< \sum_{t\geq 3} \lambda^{2t} e^{-\lambda t/100} \\
&= o(1).
\end{aligned}$$

This completes the proof of the lemma and the proof of Theorem 1. □

3. Dense random graphs. Assume now that $0 < p < 1$ is a constant. We remind the reader that $b = 1/(1-p)$.

Let

$$k = \left\lceil \left(\frac{2}{3} - \epsilon\right) \log_b n \right\rceil,$$

where $0 < \epsilon < 2/3$ is a constant. We will prove that whp $\chi_s(G(n,p)) \leq (1 + o(1))|E(G)|/k$.

Let $s = \log^2 n$, $n_0 = n/s$. Fix a partition of the vertex set $V(G)$ into s parts $V_1, \dots, V_s$ of nearly equal size: $|V_i| \approx n/s$. It will be enough to prove the following statement.

Lemma 6. *With high probability $G = G(n,p)$ satisfies the following:*

(1) *For all pairs $1 \leq i \neq j \leq s$, all but at most $O(n^2/\log^6 n)$ edges of the bipartite graph $G[V_i, V_j]$ can be packed into induced matchings of size k.*

(2) *For all pairs $1 \leq i \neq j \leq s$, $|E(V_i, V_j)| \leq n_0^2 p + n_0^{3/4}$.*

Indeed, assume that the conditions stated in the above lemma hold for G. Then we strongly color $E(G)$ as follows:

1. First, for each pair $1 \leq i \neq j \leq s$, color all but at most $n^2/\log^6 n$ edges between V_i and V_j in at most $\frac{|E(V_i,V_j)|}{k} \leq \frac{n_0^2 p}{k} + \frac{n_0^{3/4}}{k}$ colors.
2. For each $1 \leq i \neq j \leq s$, color the uncolored edges between V_i and V_j in a new color each. The total number of additional colors used for all pairs (V_i, V_j) does not exceed $\binom{s}{2}\frac{n^2}{\log^6 n} \leq n^2/\log^2 n$.
3. Color all of the edges inside each V_i in a new color. This stage consumes at most $s\binom{n/s}{2} \leq n^2/\log^2 n$ colors.

Altogether, we will have used $(1 + o(1))n^2 p/(2k)$ colors as required.

Proof of Lemma 6. Part (2) follows immediately from applying the Chernoff bounds for the binomial distribution to the number of edges joining V_i, V_j. We can thus concentrate on the bipartite graphs $G[V_i, V_j]$. Obviously, coloring such a bipartite graph is affected only by the edges between V_i and V_j and also the edges inside V_i and V_j (we are after the strong chromatic index here).

We first expose the edges of the random graph $G(n,p)$ inside the sets V_i, $1 \leq i \leq s$. Let $t = n^{2/3}$. We will be able to assume that the following two properties hold inside each V_i.

Lemma 7. *With high probability in $G = G(n,p)$, for each set V_i we have the following:*

(1) *For every collection of k disjoint sets $W_1, W_2, \dots, W_k \subset V_i$ of size $|W_i| = \nu_0 \geq n^{1/3}/\log^6 n$ there is an independent transversal in G, i.e., an independent set of vertices $\{w_1, w_2, \dots, w_k\}$ such that $w_i \in W_i, i = 1, 2, \dots, k$.*

(2) *V_i contains a collection $\mathcal{I}_i$ of $O(n^{5/3}/\log n)$ independent sets of size k in G such that $|I_{l_1} \cap I_{l_2}| \leq 1$ for each $I_{l_1} \neq I_{l_2} \in \mathcal{I}_i$, and each vertex $v \in V_i$ participates in $t(v)$ sets from $\mathcal{I}_i$, where $t(v) \in [t \pm n^{5/9}]$.*

Proof. For (1), fix $W_1, W_2, \dots, W_k \subset V_i$ and let X be the number of independent transversals. Then

$$\mathbf{E}(X) = \nu_0^k (1-p)^{\binom{k}{2}}.$$

We can now apply Janson's inequality; see, for example, Janson, Łuczak, and Ruciński [4]. Thus let

$$\begin{aligned}\Delta &= \sum_{l=2}^{k} \binom{k}{l} \nu_0^{2k-l} (1-p)^{2\binom{k}{2} - \binom{l}{2}} \\ &\leq k^2 \nu_0^{2k-2} (1-p)^{2\binom{k}{2} - 1}.\end{aligned}$$

The last inequality follows from the fact that the sum is dominated by the term $l = 2$.

Indeed, the ratio of the lth to the second term is

$$\frac{\binom{k}{l}\nu_0^{2k-l}(1-p)^{2\binom{k}{2}-\binom{l}{2}}}{\binom{k}{2}\nu_0^{2k-2}(1-p)^{2\binom{k}{2}-1}} \le \frac{k^{l-2}}{\nu_0^{l-2}(1-p)^{\binom{l}{2}-1}} = \left(\frac{k}{\nu_0(1-p)^{(l+1)/2}}\right)^{l-2} \le n^{-\epsilon(l-2)/3}.$$

Janson's inequality implies that

$$\begin{aligned}\mathbf{Pr}(X=0) &\le \exp\left\{-\frac{\mathbf{E}(X)^2}{2\Delta}\right\}\\ &\le \exp\left\{-\frac{\nu_0^2(1-p)}{k^2}\right\}.\end{aligned}$$

The number of choices for $i, W_1, W_2, \dots, W_k$ is certainly less than $n^{k\nu_0}$, and so the probability that there exists a collection without an independent transversal is at most

$$n^{k\nu_0}\exp\left\{-\frac{\nu_0^2(1-p)}{k^2}\right\} = o(1).$$

To get (2), we can argue as follows. Observe that whp every set of $1 \le j \le k$ vertices of V_i has $(1+O(n^{-1/6}))n_0(1-p)^j$ common nonneighbors in V_i. Indeed, by the Chernoff bound the probability that there is a set S, $|S| = j \le k$ for which the number of common nonneighbors lies outside $[(1\pm\theta)n_0(1-p)^j]$ $(\theta = n^{-1/6})$ is at most

$$2\sum_{j=1}^{k}\binom{n}{j}\exp\{-\theta^2 n_0(1-p)^j/3\} \le 2\sum_{j=1}^{k}\binom{n}{j}e^{-n^{\varepsilon}/(3\log^2 n)} = o(1).$$

This enables us to conclude that whp the number of independent sets $\tau(v)$ of size k contained in V_i and containing vertex $v \in V_i$ is asymptotically equal to $\mu = \binom{n_0-1}{k-1}(1-p)^{\binom{k}{2}}$. Indeed, given the above property, it follows by induction on $j \le k$ that for all $v \in V_i$ there are between $(1-jn^{-1/6})\binom{n_0-1}{j-1}(1-p)^{\binom{j}{2}}$ and $(1+jn^{-1/6})\binom{n_0-1}{j-1}(1-p)^{\binom{j}{2}}$ independent sets of size j in V_i which contain v, i.e., $|\tau(v)-\mu| \le n^{-1/6+o(1)}\mu$. Furthermore, we can also deduce that for a fixed pair $u, v \in V_i$ the number $\tau(u,v)$ of independent sets of size k containing both u and v will be at most $(1+\theta)^k\binom{n_0-2}{k-2}(1-p)^{\binom{k}{2}-1} = O(k\mu/n_0)$.

Form a random subfamily $\mathcal{I}_i^0$ of independent sets of size k in V_i by choosing each of them independently with probability t/μ. Then, by the Chernoff bound, for every $v \in V_i$ qs the number of elements of $\mathcal{I}_i^0$ containing v is between $\frac{\tau(v)t}{\mu} - t^{2/3}$ and $\frac{\tau(v)t}{\mu} + t^{2/3}$. Also,

$$|\mathcal{I}_i^0| = \frac{\mu n_0}{k}\cdot\frac{t}{\mu}(1+o(1)) = \frac{tn_0}{k}(1+o(1)).$$

For $u, v \in V_i$ the probability that $\mathcal{I}_i^0$ contains at least $\log n$ sets containing both u and v is at most

$$\binom{\tau(u,v)}{\log n}\left(\frac{t}{\mu}\right)^{\log n} \le \left(\frac{\tau(u,v)et}{\mu\log n}\right)^{\log n} = \left(O\left(\frac{kt}{n_0\log n}\right)\right)^{\log n} = n^{(-1/3+o(1))\log n},$$

and thus qs for each pair $u, v \in V_i$

(3.1) $\mathcal{I}_i^0$ contains at most $\log n$ sets containing u and v .

Also, observe that the number of pairs of independent sets of size k in V_i having v and another vertex in common is at most $\sum_{u\in V_i}\binom{\tau(u,v)}{2} = O(\mu^2k^2/n_0)$. Therefore the probability that $\mathcal{I}_i^0$ contains at least $n^{1/2}$ disjoint pairs of independent sets containing v and having another vertex in common is at most

$$\binom{O(\mu^2k^2/n_0)}{n^{1/2}}\left(\frac{t}{\mu}\right)^{2n^{1/2}} = \left(O\left(\frac{\mu^2k^2}{n_0^{3/2}}\cdot\frac{t^2}{\mu^2}\right)\right)^{n^{1/2}} = n^{-(1/6-o(1))n^{1/2}} .$$

(Pairs $(I_1, I_2), (I_3, I_4)$ are disjoint if $\{I_1, I_2\} \cap \{I_3, I_4\} = \emptyset$.) We conclude that qs for all $v \in V_i$

(3.2) $\mathcal{I}_i^0$ contains at most $n^{1/2}$ disjoint pairs of sets sharing v and another vertex.

Assume now that properties (3.1) and (3.2) hold. Then we claim that for every $v \in V_i$, $\mathcal{I}_i^0$ contains at most $n^{1/2}\log^2 n$ pairs of independent sets sharing v and another vertex. Suppose this is not so. Observe that by property (3.1) each independent set $I \in \mathcal{I}_i^0$, containing v, has another vertex in common with at most $|I|\log n < \log^2 n - 1$ sets from $\mathcal{I}_i^0$ containing v. Therefore if we form a maximal by inclusion family of disjoint pairs in $\mathcal{I}_i^0$ containing v and sharing another vertex, its size will be more than $n^{1/2}\log^2 n/\log^2 n = n^{1/2}$—a contradiction to property (3.2).

Deleting from $\mathcal{I}_i^0$ one independent set from each pair of sets sharing more than one vertex, we obtain a family $\mathcal{I}_i$ with $O(tn_0/k) = O(n^{5/3}/\log n)$ sets, in which each pair of sets has at most one vertex in common, and every $v \in V_i$ belongs to $t(v)$ sets from $\mathcal{I}_i$, where $t(v) \in [t - t/n^{1/7}, t + t/n^{1/7}]$. □

From now on we assume that the conditions stated in Lemma 7 hold. Let us concentrate on the pair (V_i, V_j), $i < j$.

Let $v \in V_i$. Assume that $(u, v) \in E(G)$, where $u \in V_j$. We define

$$R(v, u) = \{I_l \in \mathcal{I}_i : v \in I_l, N(u) \cap I_l = \{v\}\};$$

i.e., I_l is in $R(v, u)$ iff $v \in I_l$ and v is the only neighbor of u in I_l. Before diving into technical details, let us explain the main idea of the proof. Let $I_l = \{v_1, \ldots, v_k\}$. Assume that $u_1, \ldots, u_k$ form an independent set of size k in V_j such that $(v_i, u_i) \in E(G)$ and $I_l \in R(v_i, u_i)$ for all $1 \le i \le k$. Then the set of edges $\{(v_i, u_i) : 1 \le i \le k\}$ forms an induced matching of size k in G. Our aim will be to pack most of the edges between V_i and V_j in such matchings. To this end, we assign each edge (v, u) of G between V_i and V_j to one of the independent sets $I_l \in R(v, u)$. Then, for each set $I_l \in \mathcal{I}_i$ we distribute almost all the edges assigned to I_l between induced matchings of size k as indicated above.

Assume $e = (v, u) \in E(G)$ for $v \in V_i$, $u \in V_j$. We assign edge e to one of the independent sets containing v as follows: If $R(v, u) = \emptyset$, then e stays unassigned; otherwise e is assigned to a random member of $R(v, u)$. Denote

$$\rho = (1 - p)^{k-1} = n^{-2/3+\epsilon+o(1)} .$$

Recall that we denoted by $t(v)$ the number of independent sets in $\mathcal{I}_i$ containing v. The probability that e stays unassigned, conditioned on $e \in E(G)$, is $(1-\rho)^{t(v)} \le e^{-n^{\epsilon+o(1)}}$. Therefore applying the union bound we can conclude that qs every edge $e = (u, v)$ of every pair (V_i, V_j) gets assigned. Also,

$$\begin{aligned}\mathbf{Pr}\big[e \text{ gets assigned to } I_l\big] &= \sum_{r=1}^{t(v)} \frac{1}{r}\binom{t(v)-1}{r-1}\rho^r(1-\rho)^{t(v)-r} \\ &= \frac{1}{t(v)}\left(1-(1-\rho)^{t(v)}\right).\end{aligned}$$

(The parameter r above counts the number of independent sets in $R(v,u)$.)

Now let $I_l \in \mathcal{I}_i$, $v \in I_l$. Denote by $T(v,l)$ the set of neighbors u of v in V_j such that (v,u) is assigned to I_l. Note that for any two edges $e=(v,u)$ and $e'=(v,u')$ the events e and e' *gets assigned to* I_l depend on disjoint sets of pairs of vertices in $G(n,p)$ and are mutually independent. Hence, conditioned on the degree $d(v,V_j)$, the random variable $|T(v,l)|$ is distributed binomially with parameters $d(v,V_j)$ and $\frac{1}{t(v)}(1-(1-\rho)^{t(v)})$. The degree $d(v,V_j)$ in turn is also distributed binomially with parameters $n_0 = |V_j|$ and p. So, applying the Chernoff bound twice, we can argue that whp

$$\big||T(v,l)| - n_0p/t\big| \le n^{3/10} \tag{3.3}$$

for all (V_i,V_j), $I_l \in \mathcal{I}_i$, $v \in I_l$.

Now consider $I_l \in \mathcal{I}_i$. Assume $I_l = \{v_1,\dots,v_k\}$. The sets $T(v_i,l)$ are pairwise disjoint by construction. As long as $|T(v_i,l)| \ge n^{1/3}/\log^6 n$, we repeat the following procedure:

1. Find an independent transversal for the family $\{T(v_i,l)\}_{i=1}^k$. Let it be $(u_1,\dots,u_k)$, where $u_i \in T(v_i,l)$. This is possible due to the first condition of Lemma 7.
2. The set of edges $M = \{(v_1,u_1),\dots,(v_k,u_k)\}$ forms an induced matching. We color M by a fresh color.
3. Update $T(v_i,l) := T(v_i,l) - u_i$ for $1 \le i \le k$.

When this process stops, $|T(v_i,l)| \le n^{1/3}/\log^6 n + 2n^{3/10}$, due to (3.3), and hence

$$\left|\bigcup_{i=1}^k T(v_i,l)\right| \le \left(\frac{n^{1/3}}{\log^6 n} + 2n^{3/10}\right)k = O\left(\frac{n^{1/3}}{\log^5 n}\right).$$

Since whp every edge is assigned to some $T(v,l)$, altogether the number of edges between V_i and V_j that are left uncolored is at most

$$|\mathcal{I}_i| \cdot O\left(\frac{n^{1/3}}{\log^5 n}\right) = O\left(\frac{n^2}{\log^6 n}\right).$$

This completes the proof of Lemma 6 and thus of Theorem 2. □

4. Concluding remarks.

- We strongly believe that the bound we obtain here for the dense case can be further improved, and that in fact the following holds true (it was first conjectured in [2]).

  CONJECTURE 8. *Let $p=p(n)$ satisfy $n^{-1+\epsilon} \le p(n) \le 0.99$, where $0<\epsilon<1$ is a constant. Then whp in the random graph $G=G(n,p)$,*

  $$\chi_s(G) = (1+o(1))\frac{n^2p}{2\log_b n},$$

  where $b = 1/(1-p)$.

Proving this conjecture, even for a single value of $p(n)$, seems to be quite a challenging task. It appears that the proof method employed in the current paper has exhausted its potential, and new ideas are needed to establish the above conjecture.

- We prove that for random graph $G = G(n, p)$, where $np \ll \sqrt{\log n/\log\log n}$, whp $\chi_s(G) = \Delta_1(G)$. A simple first moment calculation shows that this is no longer true when $np \gg \sqrt{\log n}$. Hence the range of p in the assertion of Theorem 1 is not very far from being best possible. Nevertheless, it would be interesting to determine or at least to estimate the edge probability threshold at which the equality $\chi_s(G) = \Delta_1(G)$ ceases to be valid.

REFERENCES

[1] B. Bollobás, *Random Graphs*, 2nd ed., Cambridge Stud. Adv. Math. 73, Cambridge University Press, Cambridge, UK, 2001.

[2] A. Czygrinow and B. Nagle, *Bounding the strong chromatic index of dense random graphs*, Discrete Math., 281 (2004), pp. 129–136.

[3] R. J. Faudree, A. Gyárfás, R. H. Schelp, and Zs. Tuza, *Induced matchings in bipartite graphs*, Discrete Math., 18 (1989), pp. 83–87.

[4] S. Janson, T. Łuczak, and A. Ruciński, *Random Graphs*, John Wiley and Sons, New York, 2000.

[5] M. Molloy and B. Reed, *A bound on the strong chromatic index of a graph*, J. Combin. Theory Ser. B, 69 (1997), pp. 103–109.

[6] Z. Palka, *The strong edge colorings of a sparse random graph*, Australas. J. Combin., 18 (1998), pp. 219–226.

[7] V. H. Vu, *A general upper bound on the list chromatic number of locally sparse graphs*, Combin. Probab. Comput., 11 (2002), pp. 103–111.

SIAM J. Discrete Math.
Vol. 19, No. 3, pp. 728–743

SOLVABILITY OF GRAPH INEQUALITIES[*]

MARCUS SCHAEFER[†] AND DANIEL ŠTEFANKOVIČ[‡]

Abstract. We investigate a new type of graph inequality (in the tradition of Cvetković and Simić [*Contributions to Graph Theory and Its Applications*, Technische Hochschule Ilmenau, Ilmenau, Germany, 1977, pp. 40–56] and Capobianco [*Ann. New York Acad. Sci.*, 319 (1979), pp. 114–118]) which is based on the subgraph relation and which allows as terms fixed graphs, graph variables with specified vertices, and the operation of identifying vertices. We present a simple graph inequality that does not have a solution and show that the solvability of inequalities with only one graph variable and one specified vertex can be decided (in nondeterministic exponential time). The solvability of graph inequalities over directed graphs, however, turns out to be undecidable.

Key words. graph inequalities, graph theory, decidability

AMS subject classifications. 05C99, 03D15

DOI. 10.1137/S0895480199360655

1. A simple graph inequality.

Consider the diagram in Figure 1.

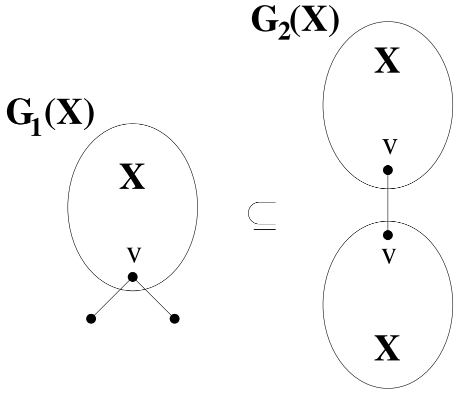


Fig. 1. *A graph inequality $G_1(X) \subseteq G_2(X)$.*

Is there a solution to this inequality? More precisely, is there an undirected graph X with a vertex v such that if we construct a graph $G_2(X)$ by taking two copies of X and connecting their v vertices by an edge, and a graph $G_1(X)$ by adding two new vertices to X and connecting them with v, then $G_1(X)$ occurs as a subgraph of $G_2(X)$?

A moment's reflection will show that the answer is yes: Take X to be a path of length two together with an isolated vertex v. What happens if we restrict ourselves to connected graphs? Again the answer is yes: Take a rooted infinite ternary tree and connect its root by an edge to a new vertex v. What about finite and connected

[*] Received by the editors August 19, 1999; accepted for publication (in revised form) January 5, 2005; published electronically December 7, 2005.
http://www.siam.org/journals/sidma/19-3/36065.html

[†] School of CTI, DePaul University, 243 S. Wabash Ave., Chicago, IL 60604 (mschaefer@cs.depaul.edu).

[‡] Department of Computer Science, University of Chicago, 1100 East 58th St., Chicago, IL 60637 (stefanko@cs.uchicago.edu).

graphs? the patient reader will ask. The answer in this case is no, there is no finite, connected graph X fulfilling the inequality in Figure 1, and this is the main result of this section.

THEOREM 1.1. *There is no connected, finite solution of the Figure* 1 *inequality.*

A simpler version of this theorem (for finite trees) was used in the first author's thesis [Sch99, Sch01] to determine the computational complexity of the arrowing relation in graph Ramsey theory: deciding $F \to (T, K_n)$ is complete for the second level of the polynomial-time hierarchy (where F is a finite graph, T is a finite tree of size at least two, and K_n is the complete graph on n vertices).

Graph equations (more so than graph inequalities) have been studied for a while, and there are two survey papers dating back to the late 1970s [CS77, CS79, Cap79]. The equalities and inequalities considered in these papers are more general in that they allow arbitrary operations on graphs such as complementation, tensor products, and squaring. Capobianco, Losi, and Riley, for example, showed that there are no (nontrivial) trees whose square is the same as their complement [CLR89]. The more general question of which graphs fulfill $G^2 = \overline{G}$ is still open [BST94], but it is known that the equation has infinitely many solutions [CK95].

We conclude this section with a proof of Theorem 1.1. Section 2 contains a generalization of this result: The solvability of graph inequalities with only one variable having one specified vertex can be decided. In section 3 we show that a natural generalization of graph inequalities leads to an undecidable solvability problem. Section 4 contains stronger results for graph inequalities over directed graphs: While the solvability of directed graph inequalities with only one variable and one specified vertex remains decidable, we can show that the solvability of directed graph inequalities is undecidable (even with at most three variables and two specified vertices for each variable).

Before we begin the proof we introduce some standard notation [Die97]. We write $G = (V, E)$ for a graph G with *vertex set* $V = V(G)$ and an *edge set* $E = E(G)$. The edge between vertices $u, v \in V$ is written as (u, v). The *order* of a graph is defined as $|V(G)|$, and the *size* $|G|$ is defined as $|E(G)|$. A graph is *finite* if it has finite order and *connected* if there is a path between any two of its vertices.

Proof of Theorem 1.1. Let X be a minimal solution of the inequality. Denote the copies of X in $G_2(X)$ by X_i, $i = 1, 2$. An *element* of X is either its edge or vertex. Given an element x of X, we denote the corresponding element of X_i by x_i.

Let ϕ be the embedding of $G_1(X)$ into $G_2(X)$. Clearly $(v_1, v_2) \in \text{Im } \phi$, since otherwise $G_1(X)$ would map into X_1 or X_2. Assume that there is an edge $e \in X$ such that neither e_1 nor e_2 is in $\text{Im } \phi$. Let Y be the connected component of $X - \{e\}$ containing v. From the connectedness of $G_1(X)$ it follows that $\text{Im } \phi \subseteq G_2(Y)$. Now the restriction of ϕ to $G_1(Y)$ is an embedding of $G_1(Y)$ into $G_2(Y)$, contradicting the minimality of X.

Thus for every $e \in X$ either e_1 or e_2 is in $\text{Im } \phi$. Note that this implies that for every vertex $u \in X$ either u_1 or u_2 is in $\text{Im } \phi$. Let Y_i be the subgraph of X corresponding to $\text{Im } \phi \cap X_i$ (as a subgraph of X_i). Then for each $e \in X$ either $e \in Y_1$ or $e \in Y_2$. We know that

$$Y_1 \cup Y_2 = X, \tag{1}$$

$$|V(Y_1)| + |V(Y_2)| = |V(\text{Im } \phi)| = |V(G_1(X))| = |V(X)| + 2, \tag{2}$$

$$|E(Y_1)| + |E(Y_2)| = |E(\text{Im } \phi)| - 1 = |E(G_1(X))| - 1 = |E(X)| + 1. \tag{3}$$

The first equality in (3) follows from the fact that $(v_1, v_2) \in \text{Im } \phi$, but $(v_1, v_2) \notin \text{Im } \phi \cap$

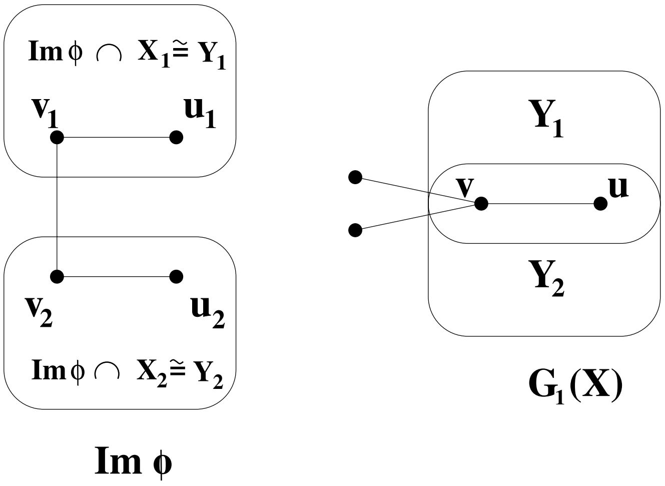


FIG. 2. *Im ϕ and $G_1(X)$.*

$(X_1 \cup X_2)$. From (1), (2), (3) we conclude that $|V(Y_1 \cap Y_2)| = 2$ and $|E(Y_1 \cap Y_2)| = 1$, which implies that the intersection of Y_1 and Y_2 is a single edge f. We know that $v \in V(Y_1) \cap V(Y_2)$, and hence $f = (v, u)$ for some $u \in V(X)$. Figure 2 illustrates the situation.

Let a_i be the number of vertices from $V(Y_i) \setminus \{u, v\}$ which have degree 1 in X. Let b be 1 if u has degree 1 in X and 0 otherwise. The number of vertices of degree 1 in $G_1(X)$ is $a_1 + a_2 + b + 2$. The number of vertices of degree 1 in Im ϕ is at most $a_1 + a_2 + b + 1$. Hence Im ϕ and $G_1(X)$ are not isomorphic, a contradiction. □

2. Decidability of graph inequalities. We could now start considering all kinds of diagrams involving graphs, vertices, edges, and the subgraph relationship. How hard is it to settle these questions? In this section we will show that the solvability of graph inequalities of the type presented in the previous section, i.e., having only one graph variable with one specified vertex, is decidable. This will follow from an (exponential) upper bound on the size of a minimal solution (if there is one). This result will be complemented by the undecidability result of the next section.

Let us formalize the question. A graph variable X with a set of specified vertices $v_1, \dots, v_m$ represents an unknown finite, connected graph whose vertex set includes vertices $v_1, \dots, v_m$. Given several graph variables $X_1, \dots, X_n$ and a graph G, we can construct a graph term $G(X_1, \dots, X_n)$ (called *gterm*) by taking several copies of each X_i and identifying some specified vertices of the copies with some vertices of G. Since we are working with connected graphs we require $G(X_1, \dots, X_n)$ to be connected (for any assignment of connected graphs to $X_1, \dots, X_n$). Note that G itself does not have to be connected and that if $G(X_1, \dots, X_n)$ is connected for some assignment of connected graphs to $X_1, \dots, X_n$, then it is connected for all assignments.

Given two such gterms $G_1(X_1, \dots, X_n)$, $G_2(X_1, \dots, X_n)$, we can ask whether there exists an assignment of connected finite graphs to the variables $X_1, \dots, X_n$ such that $G_1(X_1, \dots, X_n)$ is a subgraph of $G_2(X_1, \dots, X_n)$. We call a question of this type a *graph inequality.*

For the rest of this section we will consider the simplest possible case of a graph inequality: only one variable, X, with one specified vertex v. Let $G_1(X)$ be a gterm

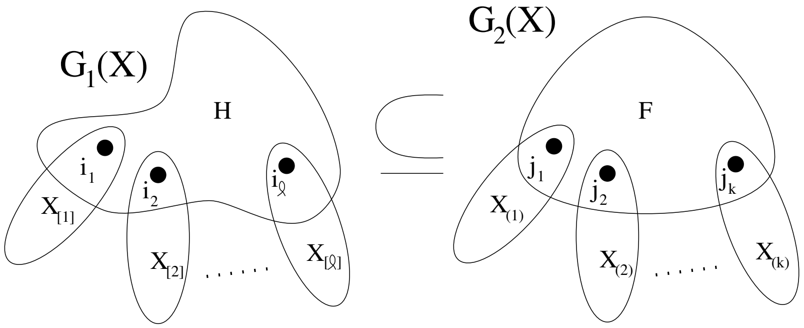


FIG. 3. *Inequality* $G_1(X) \subseteq G_2(X)$.

consisting of a connected graph H and a copy of X attached with v to each vertex of a multisubset $I = \{i_1, \dots, i_\ell\}$ of vertices of H. Similarly construct $G_2(X)$ from a connected graph F and a multisubset $J = \{j_1, \dots, j_k\}$ of vertices of F. The copy of X in $G_2(X)$ attached to j_r $(1 \le r \le k)$ is called $X_{(r)}$, and the copy of X in $G_1(X)$ attached to i_r $(1 \le r \le \ell)$ is called $X_{[r]}$. If there is only one copy of X in $G_1(X)$, we call it X.

THEOREM 2.1. *If the inequality in Figure* 3 *has a solution* X, *then it has a solution of size at most* $|F|(1+k)^{|H|}$.

The upper bound on the size of a minimal solution is exponential in the size of the equality; hence to decide solvability we just have to test all graphs up to that size, something which can be done in nondeterministic exponential time (NEXP).

COROLLARY 2.2. *The solvability of graph inequalities of the type in Figure* 3 *can be decided in* NEXP.

We do not know the precise computational complexity of the decision problem. It is at least NP-hard, since we can ask whether a graph contains a clique.

At the core of the proof are Lemmas 2.5 and 2.7, which show that for a minimal solution to the graph inequality (if it exists) we can assume that all of the vertices of I are mapped to vertices of F. This reduces the problem to a simpler variant (namely, the images of vertices from I are prescribed) dealt with by Lemma 2.4 (based on the representation result of Lemma 2.3).

First we characterize solutions of inequalities (with prescribed mapping) where on the left-hand side there is only one copy of X and v has to map to a vertex w of F on the right-hand side.

If $w \in J$, then any connected graph is a solution. Now assume $w \notin J$. Let Σ be the alphabet consisting of the numbers $1, \dots, k$. For each word α from Σ^* take a copy $F^{(\alpha)}$ of F. For every $\alpha \in \Sigma^*$ and $a \in \Sigma$ identify $w^{(\alpha a)}$ and $j_a^{(\alpha)}$. The resulting infinite graph is called F^∞ (see Figure 5).

LEMMA 2.3. *Assume that* $w \notin J = \{j_1, \dots, j_k\}$. *Then the solutions of the inequality in Figure* 4 *are precisely the subgraphs* X *of* F^∞ *with* $v = w^{()}$ *such that*

$$\text{for any edge } e \text{ in } F, \text{ any } \alpha \in \Sigma^*,\ a \in \Sigma, \\ \text{if the edge } e^{(a\alpha)} \text{ is in } X, \text{ then } e^{(\alpha)} \text{ is in } X. \tag{4}$$

Proof. If X is a subgraph of F^∞ satisfying condition (4), then X is a solution of the inequality via mapping ϕ:

$$\phi(x^{()}) = x,$$

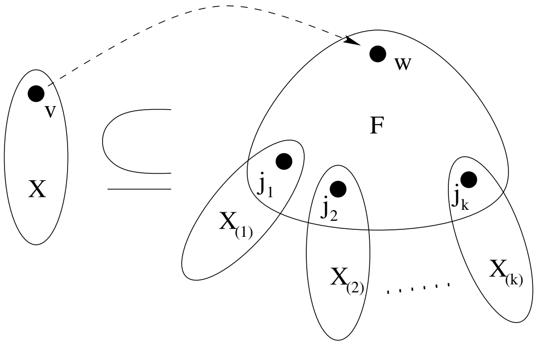


FIG. 4. *Inequality* $X \subseteq G_2(X),\ v \to w$.

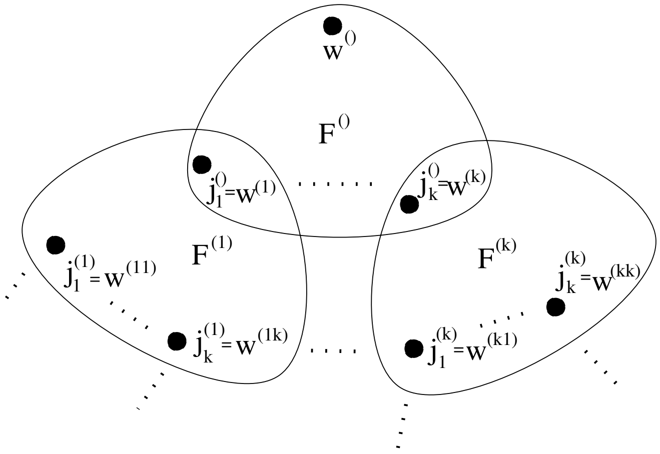


FIG. 5. F^∞.

$$\phi(x^{(a\alpha)}) = x^{(\alpha)}_{(a)}.$$

If X is a solution of the inequality via mapping $\phi : X \to G_2(X)$, then define

$$Y^{()} = \phi^{-1}(F),$$
$$Y^{(a\alpha)} = \phi^{-1}(Y^{(\alpha)}_{(a)}),$$

where $Y^{(\alpha)}_{(a)}$ is the copy of $Y^{(\alpha)}$ in $X_{(a)}$ in $G_2(X)$. If e is an edge of X with distance d from v, then it must map either to F or to some edge f in some $X_{(r)}$ which has strictly smaller distance from $v_{(r)}$ than d. Edges adjacent to v must be mapped to F, and hence they are in $Y^{()}$. By induction it follows that

$$X = \bigcup_{\alpha \in \Sigma^*} Y^{(\alpha)}.$$

Clearly $Y^{(\alpha)}$ is a subgraph of F via $\phi^{|\alpha|+1}$ for any $\alpha \in \Sigma^*$. The element of $Y^{(\alpha)}$ corresponding to $x \in F$ is called $x^{(\alpha)}$. By induction it follows that $w^{(\alpha a)} = j_a^{(\alpha)}$ for any $\alpha \in \Sigma^*,\ a \in \Sigma$. From the definition of Y's, if $e^{(a\alpha)}$ is in X, then the edge $e^{(\alpha)}$ is also in X for any $\alpha \in \Sigma^*,\ a \in \Sigma$. Hence X is a subgraph of F^∞ satisfying (4). □

Solving systems of simple graph inequalities is useful in solving more complicated inequalities.

LEMMA 2.4. *If a system of inequalities with prescribed mappings*

$$H_1 \subseteq X,\ h_1 \to v; \ldots ; H_m \subseteq X,\ h_m \to v, \tag{5}$$

$$X \subseteq F_1(X),\ v \to w_1; \ldots ; X \subseteq F_n(X),\ v \to w_n \tag{6}$$

has a solution, then it has a solution of size at most $|F_1|(1+k_1)^M$*, where* k_1 *is the number of copies of* X *in* F_1 *and* $M := \max\{|H_1|, \ldots, |H_m|\}$*, assuming that the graphs* $H_1, \ldots, H_m$ *are connected.*

Proof. Let X be a minimal solution of the system. Let e be an edge of X whose distance d from v is maximal. Assume that $d > M$. If we remove the edge e, then $X' = X - \{e\}$ still satisfies inequalities (5), because no edge of any H_i $(1 \le i \le m)$ can map to e. If X satisfies the inequality in Figure 4 for $F = F_i$ $(1 \le i \le n)$, then by Lemma 2.3 it is a subgraph of F^∞ with $v = w^{()}$ and it satisfies condition (4). Let $e = f^{(\alpha)}$. Clearly X' is also a subgraph of F^∞ and the condition is still satisfied, because $\mathrm{dist}(v, f^{(a\alpha)}) > \mathrm{dist}(v, f^{(\alpha)})$ and hence $f^{(a\alpha)} \notin X'$ for any $a \in \Sigma$. Therefore X' satisfies inequalities (6), a contradiction to the minimality of X.

Thus $\mathrm{dist}(v, e) \le M$. The size of the subgraph of F_1^∞ consisting of edges within distance M from v is bounded by $|F_1|(1+k_1)^M$. □

Now we return to the inequality in Figure 3.

LEMMA 2.5. *If there is more than one copy of* X *on the left side of the inequality in Figure* 3*, then every* $i_r = v_{[r]}$ $(1 \le r \le \ell)$ *must map to a vertex of* F*.*

Proof. Suppose, for example, that i_1 maps into some $X_{(r)} - \{j_r\}$. Let P be a path from i_1 to i_2. Graphs $X_{[1]}$ and $X_{[2]} \cup P$ share only vertex i_1. Hence the image of at least one of them does not contain j_r and since j_r is a cutvertex of G_2, that image must be contained in $X_{(r)} - \{j_r\}$, which is impossible, since there are more vertices in X_1 or in $X_2 \cup P$ than in $X_{(r)} - \{j_r\}$. □

LEMMA 2.6. *If* X *is a solution of the inequality in Figure* 3 *via mapping* $\psi : G_1(X) \to G_2(X)$*, then there exists a mapping* $\phi : G_1(X) \to G_2(X)$ *such that* $\phi(i) = \psi(i)$ *and as many copies of* X *in* i *as possible are mapped to copies of* X *in* $\phi(i)$ *for every* $i \in I$*.*

Proof. Consider a bipartite graph B with partitions I and J, where i_r is connected to j_s if and only if $\psi(i_r) = j_s$. Without loss of generality assume that $\{(i_r, j_r); 1 \le r \le t\}$ is a maximal matching of B.

We need to show that there exists ϕ such that $X_{[r]}$ maps to $X_{(r)}$ for $1 \le r \le t$. Let $Y^1, \ldots, Y^q$ be the connected components of $X - \{v\}$. Let ϕ be a mapping such that

$$\sum_{r=1}^{t} \sum_{j=1}^{q} \left| \phi(Y^j_{[r]}) \cap Y^j_{(r)} \right| \tag{7}$$

is maximal. If for some r, j,

$$\phi\big(Y^j_{[r]}\big) \neq Y^j_{(r)},$$

then clearly $\phi(Y^j_{[r]}) \cap Y^j_{(r)} = \emptyset$; otherwise $\phi(Y^j_{[r]})$ would have to contain j_r. Now we can change ϕ in such a way that $Y^j_{[r]}$ will be mapped to $Y^j_{(r)}$ and $\phi^{-1}(Y^j_{(r)})$ will be mapped to $\phi(Y^j_{[r]})$. This increases the value of (7), a contradiction. Hence ϕ maps $X_{[r]}$ to $X_{(r)}$ $(1 \le r \le t)$. □

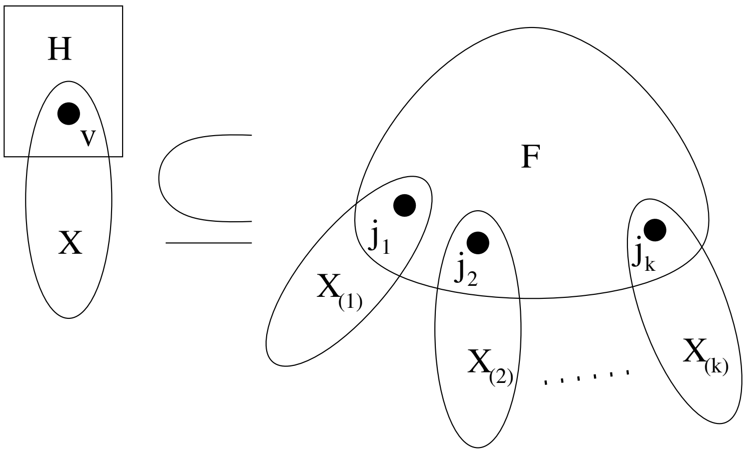


FIG. 6.

We prove an analogue of Lemma 2.5 for inequalities where X occurs only once on the left-hand side of the inequality in Figure 3.

LEMMA 2.7. *If the inequality in Figure* 6 *has a solution, then it has a solution* X *via a mapping* ϕ *which maps* $v = i_1$ *to a vertex of* F.

Proof. Suppose that there is no solution of the inequality in Figure 6 such that v maps to a vertex of F, but there is a solution in which v maps into a vertex of $X_{(1)} - \{j_1\}$. Then clearly the inequality in Figure 7 with the additional condition that v must map to some $u \in X_{(1)}$ has a solution (see Figure 7).

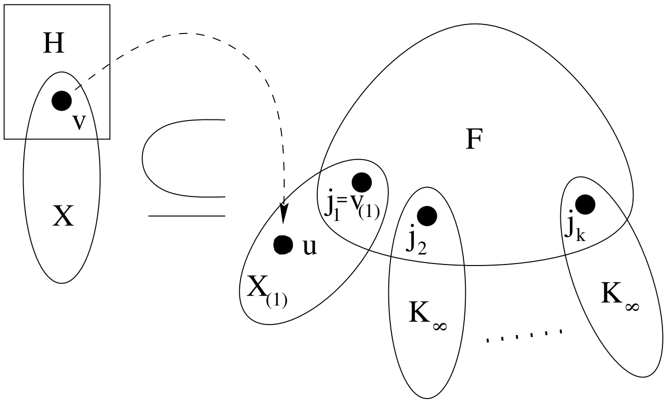


FIG. 7.

If $u = j_1$, then by Lemma 2.6 there is ϕ such that X is mapped to $X_{(1)}$. Therefore we can replace K_∞'s in the inequality in Figure 7 by $K_{|H|}$'s, since only H is mapped to $G_2(X) - X_{(1)}$. This, however, implies that $X = K_{|H|}$ is a solution of the inequality in Figure 6 in which v maps to a vertex of F, a contradiction.

Thus $u \neq j_1$ for every solution of the inequality in Figure 7. Let X be a minimal solution of this inequality. Graphs H and X share only v; moreover j_1 is a cutvertex of G_2 and hence either H or X must be mapped inside $X_{(1)} - \{j_1\}$. Since the latter is not possible, H must be mapped inside $X_{(1)} - \{j_1\}$.

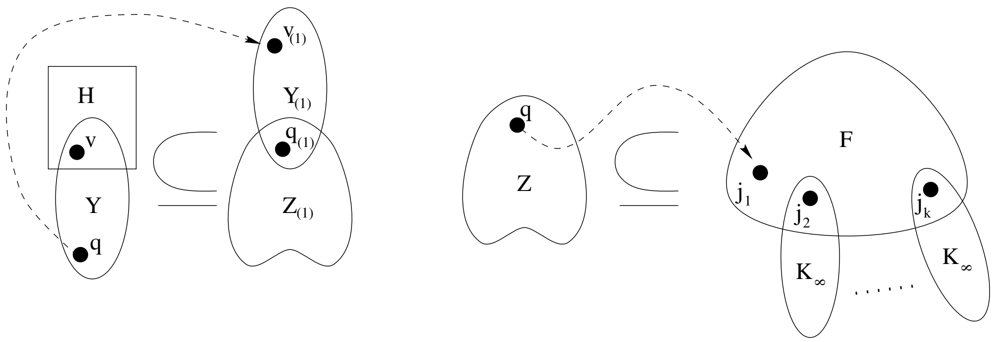


FIG. 8.

Now let $Y = \phi^{-1}\left(X_{(1)}\right) \cap X$ and $Z = \phi^{-1}\left(G_2(X) - (X_{(1)} - \{j_1\})\right)$. The common vertex of Y and Z is called $q = \phi^{-1}(j_1)$. The inequality in Figure 7 implies the inequalities in Figure 8.

The second inequality follows directly from the definition. To see the first inequality, note that the graph on the left-hand side is a subgraph of $X_{(1)}$ with q mapping to $j_{(1)}$, and that by definition of Y and Z the right-hand side contains $X_{(1)}$ with $j_{(1)}$ of $X_{(1)}$ mapping to v of Y.

If in the first inequality v was mapped outside of $Y_{(1)}$, then the shortest path from q to v would have to map to a longer path, which is not possible. Hence v maps inside $Y^{(1)}$. Combining the two inequalities in Figure 8, we get that Y satisfies the inequality in Figure 7. This contradicts the minimality of X. □

We can now complete the proof of Theorem 2.1 by showing a bound on the size of a minimal solution (if there is one) of graph inequalities with one variable and one specified vertex.

Proof of Theorem 2.1. From Lemmas 2.5 and 2.7 it follows that we need to consider only solutions in which every i_r $(1 \leq r \leq \ell)$ maps to a vertex of F. For each such mapping ϕ, using Lemma 2.6, we can assume that if $i \in I$ maps to a vertex $j \in J$, then as many copies of X in i as possible map to copies of X in j.

Let

$$G_1'(X) \subseteq G_2'(X), \quad v = i_1 \to \phi\left(i_1\right), \ldots, i_\ell \to \phi\left(i_\ell\right) \tag{8}$$

be the inequality with prescribed mappings obtained by removing those $X_{[r]}$'s and $X_{(r)}$'s which are already taken care of by Lemma 2.6. Notice that now no i_r', $(1 \leq r \leq \ell')$ maps to a j_s' $(1 \leq s \leq k')$.

Let X' be a solution of (8) with mapping ψ. If $\psi\big(X_{[r]}'\big) \cap X_{(s)}' \neq \emptyset$, then some vertex from $X_{[r]}' - \{i_r'\}$ must map to j_s'. Since j_s' is a cutvertex, no other part of $G_1'(X')$ can map to $X_{(s)}'$. If for each $X_{[r]}'$, $1 \leq r \leq \ell'$, and H we take the set of objects (edges and $X_{(s)}'$'s) to which it is mapped, then these sets are disjoint.

There are only finitely many partitions of the objects of $G_2'(X)$ into $\ell + 1$ disjoint sets. For each such partition we get a system of inequalities with prescribed mappings as in Lemma 2.4, which has a solution of size at most $|F|(1+k)^{|H|}$ (if it has one). □

Note that by using previous lemmas we can easily prove Theorem 1.1. If there was a solution of the inequality in Figure 1, then by Lemma 2.7 there is a solution such that v from $G_1(X)$ maps to one of the v's in $G_2(X)$. By looking at the degrees of v's we see that this is not possible.

We conclude this section with a technical result that allows us to combine several inequalities with prescribed mappings. This lemma will be needed in the next section.

LEMMA 2.8. *For any system of inequalities with prescribed mappings*

$$H_1 \subseteq X,\ h_1 \to v; \ldots ; H_m \subseteq X,\ h_m \to v,$$
$$X \subseteq F_1(X),\ v \to w_1; \ldots ; X \subseteq F_n(X),\ v \to w_n,$$

there is a single inequality which has the same set of solution as the system.

Proof. Consider the inequality in Figure 9.

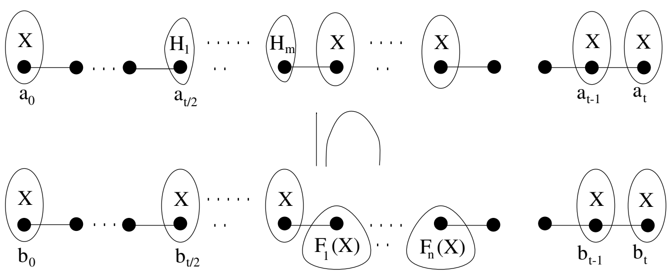


FIG. 9.

By Lemma 2.5, a_0 and a_t have to map to F. Clearly the a_0, a_t path of H in $G_1(X)$ has to map to a path in F in $G_2(X)$. If $t > 2(m + n + \max\{F_1, \ldots, F_n\})$, then the only path of length t in F is the b_0, b_t path. It follows that a_i maps to b_i $(0 \leq i \leq t)$ because a_{t-1} cannot map to a_1. Hence X is a solution of the inequality in Figure 9 if and only if it is a solution of the system. □

3. Undecidability of graph inequalities. The result of the last section might suggest that there is a general method to decide the solvability of graph inequalities. While we have to leave this question open for the time being, we do want to sketch a proof that a natural generalization of the problem turns out to be undecidable. We consider a logical language whose atoms are graph inequalities as above, i.e., diagrams involving graphs with labeled vertices, additional edges and vertices, and one occurrence of the subgraph relationship. We then build more complex formulas by allowing logical operators $\wedge$ (and) and $\neg$ (not) and quantifiers over graphs (and labeled vertices). We will not formally describe the semantics of this language since it is straightforward; the only point worth mentioning is that we assume vertices with different labels in the same graph to be different.

We will next show that formulas of this type are not decidable. More precisely we will show that this is even the case if we restrict the quantifiers in the formulas to be only existential or bounded (i.e., of the form $(\forall F \subseteq G)$ or $(\exists F \subseteq G)$). Since formulas involving only bounded quantifiers are decidable (the bounds have to be explicit graphs; hence we can try all possible combinations), this is a reasonably sharp result on the complexity of graph inequalities. The main open problem of interest, of course, is whether the problem is undecidable in case we allow only existential quantifiers (and no bounded quantifiers at all). We will mention some interesting related problems in the conclusion.

THEOREM 3.1. *The solvability of graph diagrams with Boolean operators, existential quantifiers, and bounded quantifiers is not decidable.*

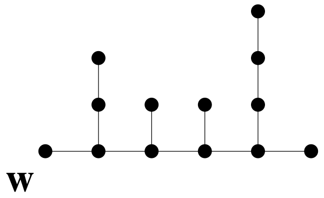


FIG. 10. *Representing the word* 21130.

Proof. We will show the undecidability of the solvability problem by reducing the word problem for semi-Thue systems to it (see, for example, [HU79]). Over an alphabet A, a semi-Thue system is a set of productions $x \Rightarrow y$ $(x, y \in A^*)$, meaning that x can be transformed into y. The word problem for a semi-Thue system is to decide whether, given two words x and y, there is a series of productions which, when applied to substrings of the words, transforms x into y.

We will represent the letters of the alphabet as paths of different lengths. A word will be coded as a path to which are attached further paths coding the letters of the word. A sequence of words will be coded in a similar way. We will then have to find a way to verify that such a sequence results from legal applications of the productions.

Fix a semi-Thue system $(x_i \Rightarrow y_i)_{i \leq n}$ over some alphabet A, and suppose we are given two words x and y. The following diagram gives an example of how we represent words, in this case the word 21130 (Figure 10).

The initial vertex w is used to link the word up in a sequence of words. In the manner depicted by the diagram we associate graphs X_i, Y_i, X, and Y with the words x_i, y_i, x, and y.

Assume that for all $A \subseteq G$ the following diagram (Figure 11) is true.

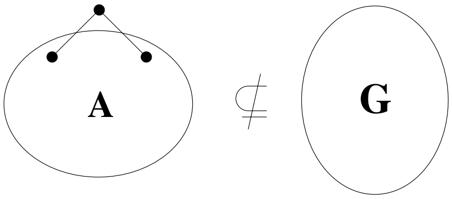


FIG. 11. *Forcing a tree.*

Then G does not contain any cycles and therefore is a tree. Furthermore, by excluding $K_{1,4}$ we can easily assure that G has maximal degree at most 3. We now set up G to code the initial and final words. We do this by saying that there is an $A \subseteq G$ which fulfills the diagram in Figure 12.

Note that for the diagram to be true w_X has to be mapped to u and w_Y to v (G is a tree). Hence G will contain a path from u to v. For each vertex w on that path let G_w be the graph attached to the path (if none, then G_w is just w). With the previous diagram we have ensured that G_{w_X} codes x and G_{w_Y} codes y. Now we have to verify only that the transitions between words as coded by G are legal according to the system of productions given. We do this by saying that for any $A, B, C, D \subseteq G$ for which the diagram in Figure 13 is true, there are $S, E, B', C' \subseteq G$ for which the diagram in Figure 14 is true, and such that $B' = X_i$ and $C' = Y_i$ for some $i \leq n$.

It is straightforward to check that in this manner we have encoded the original

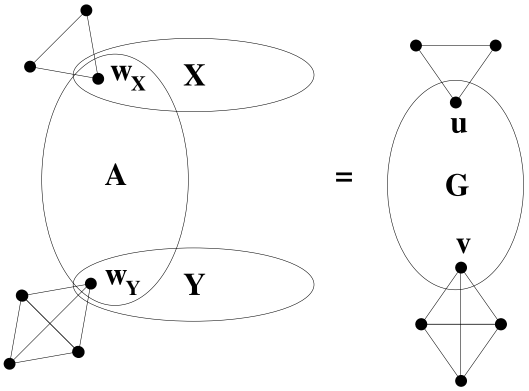


Fig. 12. *Forcing x and y.*

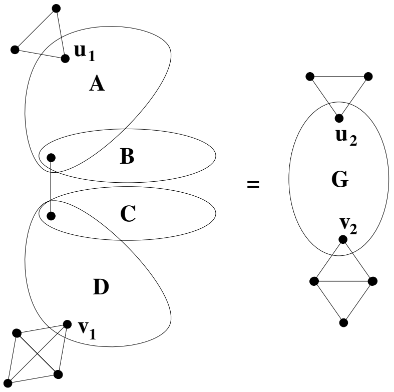


Fig. 13. *Transition from B to C.*

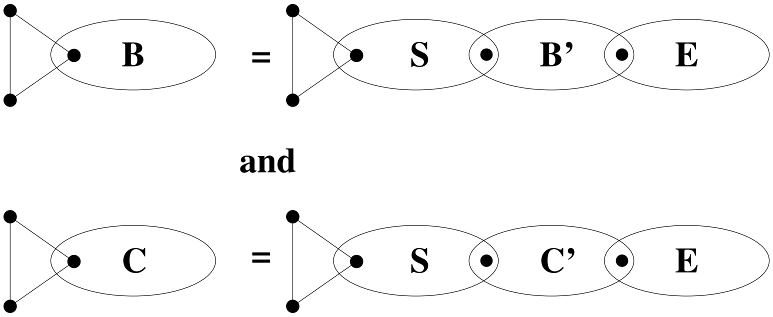


Fig. 14. *Application of production B' to C'.*

word problem: There is a G fulfilling all these conditions if and only if there is a solution to the word problem. Hence the word problem can be written as a graph inequality with one existential quantifier and some bounded quantifiers. □

4. Directed graph inequalities. So far we have considered only undirected graphs. What happens if we change the universe of graph inequalities to directed (or colored) graphs? Call these variants *directed (or colored)* graph inequalities, respectively.

In the case of one variable with one specified vertex we can obtain the same result as in Theorem 2.1. As a matter of fact, the lemmas and proofs needed for that theorem can be used without modification.

THEOREM 4.1. *For directed (or colored) graphs, if the inequality in Figure* 3 *has a solution X, then it has a solution of size at most $|F|(1+k)^{|H|}$.*

As above, this implies that the problem is decidable in NEXP.

The complexity of the undecidability proof in section 3 stemmed from the difficulty of coding the alphabet: We had to use special devices to code letters and then use bounded quantifiers to verify that the coding was correct. Allowing the edges in the graph to be directed, however, makes these constructions unnecessary.

THEOREM 4.2. *The solvability of directed (colored) graph inequalities is undecidable.*

The problem remains undecidable even if we limit it to three variables with two specified vertices each. We consider only directed graphs, since the treatment for graphs with two colors is identical.

Proof. We will translate Post's correspondence problem (PCP) into a directed graph inequality. Since the former problem is known to be Turing-complete [HU79], this shows the undecidability of directed graph inequalities.

PCP asks whether, given a list of pairs of words $(p_i, q_i)_{1 \leq i \leq n}$, there is a list of indices $i_1, \ldots, i_m$ such that $p_{i_1} \cdots p_{i_m} = q_{i_1} \cdots q_{i_m}$. PCP can be translated into a question about context-free grammars as follows: Consider two grammars

(i) $S_1 \rightarrow \underline{i}\, S_1 p_i \mid \underline{i}\, p_i \ (1 \leq i \leq n)$,

(ii) $S_2 \rightarrow \underline{i}\, S_2 q_i \mid \underline{i}\, q_i \ (1 \leq i \leq n)$,

where $\underline{i}$ is a prefix-encoding of the number i. The original problem has a solution if and only if the two grammars have a word in common, i.e., there is a word w such that $S_1 \rightarrow^* w$ and $S_2 \rightarrow^* w$.

Consider a context-free grammar with productions over the alphabet $\{0, 1\}$ and one nonterminal symbol S. Every production has S on the left-hand side and a (nonempty) string of letters and at most one occurrence of S on the right-hand side.

We will code 0's and 1's by the direction of edges, an outgoing edge coding a 0 (for a string starting in the vertex) and an incoming edge coding a 1. Let G_a be the path corresponding to the string a (for an example, see Figure 15).

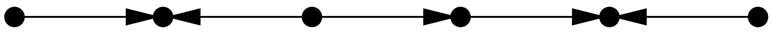


FIG. 15. G_{01001}.

A production is either of the form $S \rightarrow aSb$, where $ab \in \{0, 1\}^+$, or of the form $S \rightarrow a$, where $a \in \{0, 1\}^+$. We assume that there is always a production of the second kind.

Construct a graph inequality as follows: The left-hand side contains a graph variable X_S with two special vertices u_S and v_S. The right-hand side has two special vertices u'_S and v'_S. For every production of the form $S \rightarrow aSb$, include G_a starting in u'_S and ending in the u_S vertex of a new copy of X_S, and G_b starting in the v_S vertex of X_S and ending in v'_S. For every production of the form $S \rightarrow a$, include G_a starting in u'_S and ending in v'_S.

If we require that u and v be mapped to u' and v', respectively, then a solution to the inequality corresponds to a word in the language described by the grammar, and, vice versa, every word in the language gives rise to a solution of the graph inequality.

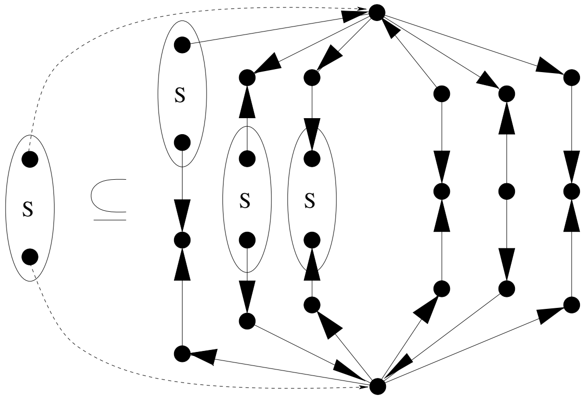


FIG. 16. *Graph inequality for semi-Thue system.*

For an example, see Figure 16, which shows the graph inequality belonging to the system $S \Rightarrow 0S100 \mid 10S11 \mid 11S00 \mid 0100 \mid 1011 \mid 1100$.

We will first prove the claim that for every word in the language there is a corresponding solution of the graph inequality in a stronger form: For each n there is a graph G_S such that

(i) G_S solves the inequality (with u, v mapping to u', v') and

(ii) there is a path G_w between u_S and v_S in G_S for every word w that can be derived in n steps from S.

We prove this statement by induction on n. For $n = 1$ let G_S consist of all paths G_a for which $S \to a$ is a production, and identify their starting vertices (calling it u_S), and their end vertices (calling it v_S). For the induction step, assume we have a graph G'_S with vertices u'_S and v'_S fulfilling the induction hypothesis for n. Build G_S with vertices u_S and v_S by including for each production $S \to aSb$ (new) copies of G_a, G'_S, and G_b, and by identifying u_S with the starting vertex of G_a, u'_S with the ending vertex of G_a, v'_S with the starting vertex of G_c, and v_S with the ending vertex of G_c. It is easy to show by induction that the graphs so constructed fulfill (i) and (ii).

For the other direction suppose that there is a solution G_S to the graph inequality. We will show that for any path P from u_S to v_S in G_S there is a word w such that $S \to^* w$ and $P = G_w$. Use induction on the length of the path: Let P be a path of minimal length between u_S and v_S for which the assertion has not yet been proven. P has length at least one (since u_S and v_S are different vertices). Fix w such that $P = G_w$. Since G_S fulfills the inequality, P must be a subpath of the right-hand side of the inequality starting in u'_S and ending in v'_S. The way the right-hand side was constructed, P must therefore be a subpath in a graph corresponding to a particular production $S \to aSb$, or $S \to a$. In the latter case, $a = w$ and we are done. In the former case, P consists of three parts corresponding to a, S, and b, respectively. Since a and b together have length at least one, we can apply the induction hypothesis to the subpath of P corresponding to S.

If we are given two grammars $\mathcal{G}_1, \mathcal{G}_2$, we can construct the inequalities for them as above and ask whether there exist graphs fulfilling them, as well as a path P from u_P to v_P which is a subgraph of both X_{S_1} and X_{S_2}, where u_P and v_P have to be mapped to u_{S_i} and v_{S_i} $(i = 1, 2)$. Such a path corresponds to a word w which can be derived in both grammars. We are left with the task of combining the inequalities

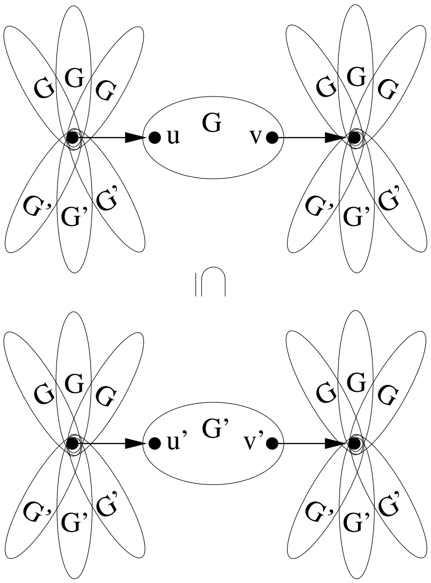


FIG. 17. $G \subseteq G'$.

into a single inequality fulfilling the additional requirements on the u and v vertices.

Consider the directed graph inequality of Figure 17.

We claim that if G and G' are solutions of this inequality, then G is a subgraph of G' such that u and v are mapped to u' and v', respectively (and, obviously, any such graphs are solutions to the inequality). To see this, suppose that one of the vertices at the heart of a sunflower does not map to its corresponding vertex. It then has to map to a labeled vertex, or into a G' or G, say G'. This is not possible, since such a vertex is at the heart of three copies of G', at most two of which can map outside the G', so there would have to be a full copy of G' within G', which is impossible. Hence the hearts of the sunflowers map to each other, and, in consequence, the copies of G map to the corresponding copies of G', while u and v map to u' and v'.

We have four equations altogether: $G_{S_i} \subseteq G_i$ (with G_i the right-hand sides constructed from the grammars) and $G_P \subseteq G_{S_i}$ $(i = 1, 2)$. We can extend the diagram above to incorporate all four inequalities: It will contain five sunflowers on each side of the inequality, between which the terms of the four inequalities are linked up; each sunflower will have three copies of each graph involved in the construction, and hence the hearts of the sunflowers map to each other, as above. Thus we get a single directed graph inequality which has a solution if and only if the two grammars have a word in common. □

5. Conclusion. Several questions remain open, the most nagging one being the complexity of deciding the solvability of (undirected) graph inequalities (without additional quantifiers and Boolean operators). It seems hard to translate the corresponding undecidability result for directed graph inequalities back to the undirected case. Another approach would be to strengthen the proof of the undirected undecidability result, which required one existential quantifier and several alternations of bounded quantifiers. It seems likely that by using a different problem for the reduction (for example, PCP) one might get the language down to existential and bounded universal

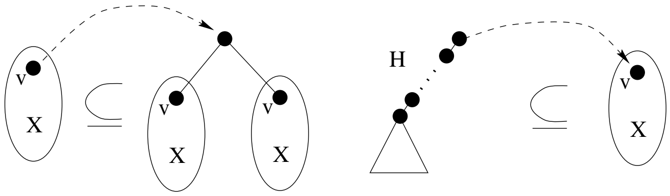


FIG. 18.

quantifiers only. Getting rid of that last layer of bounded quantifiers, thereby settling the complexity of Boolean combinations of graph inequalities, seems harder. The language shown to be undecidable in section 3, for example, is powerful enough to code the edge reconstruction conjecture (in a more or less natural fashion). Hence a decision procedure would have come as a surprise. In the case of graph inequalities the situation is different: We do not know of any difficult open problem that can be phrased as a graph inequality; hence decidability might still be an option.

Question 1. Is the solvability of graph inequalities (as defined in section 2) decidable?

A positive indication for decidability is that it seems difficult to force large solutions. If graph inequalities were undecidable, then the solution size would have to grow faster than any computable function. The best result we have been able to obtain so far shows that a quadratic lower bound is possible, a far cry from undecidability.

THEOREM 5.1. *There is a graph inequality $G_1(X) \subseteq G_2(X)$ of size $O(n)$ such that the size of a minimal solution is $\Omega(n^2)$.*

Proof. Consider the system of inequalities (Figure 18) with prescribed mappings, where H is a path of length n connected to a complete binary tree of depth $\log n$. Let B be the infinite binary tree with edges naturally labeled by strings from $\{0,1\}^+$. By Lemma 2.3 solutions of the first inequality are subgraphs X of B such that if edge $a\alpha$ is in X, then edge α is also in X for any $a \in \{0,1\}$, $\alpha \in \{0,1\}^+$. From the second inequality it follows that for any solution X there is some $\alpha \in \{0,1\}^n$ such that for every $\beta \in \{0,1\}^{\log n}$, edge $\alpha\beta$ is in X. Hence for any suffix γ of α for every $\beta \in \{0,1\}^{\log n}$, edge $\gamma\beta$ is in X and therefore there are $\Omega(n^2)$ edges in X. Using Lemma 2.8 we combine the inequalities in Figure 18 into a single inequality. □

Question 2. Are there graph inequalities whose minimal solutions have at least exponential size?

Our decidability result for graph inequalities with one variable (and one labeled vertex) shows that the computational complexity of the problem lies in NEXP. As we pointed out earlier, it is also NP-hard (since we can ask for a clique as subgraph, without even using the existential quantifier).

Question 3. What is the computational complexity of deciding the solvability of one-variable, one-vertex graph inequalities? Is the problem NEXP-complete?

First steps towards generalizations of the decidability result would probably try to increase the number of specified vertices, then the number of variables. Also, can we decide Boolean combinations of graph inequalities?

One special case of Boolean combinations can be settled with the techniques from section 2: graph equalities with one variable and one specified vertex.

THEOREM 5.2. *The solvability of graph equalities with one variable and one specified vertex is decidable.*

Proof. Lemma 2.5 allows us to assume that variable X occurs at most once on each side of the equality (otherwise we can use Lemma 2.4 as in the proof of Theorem 2.1). If X does not occur on one of the sides, we are done. If it occurs precisely once on each side, it is not too difficult to see that the equality is solvable if the two graphs to which the variable is attached are isomorphic (where the labeled vertices have to map to each other). The decision procedure outlined here is, again, in NEXP. □

In the case of directed graph inequalities we have a tight separation of decidability and undecidability: One variable with one specified vertex is decidable, and three variables with two specified vertices are not. While it might be interesting to find out what happens in the case of two variables, a more promising object of study should be the computational complexity of directed graph inequalities. The direction of the edges might help in encoding a problem complete for EXP or NEXP.

Question 4. What is the computational complexity of deciding the solvability of one-variable, one-vertex directed (or colored) graph inequalities? Is the problem NEXP-complete?

Finally we would like to suggest that the question of computational complexity should also be an interesting one for the more general types of graph equalities and graph inequalities studied in the literature [CS79].

Acknowledgments. We would like to thank Laci Babai and János Simon for helpful discussions.

REFERENCES

[BST94] V. Baltić, S. K. Simić, and V. Tintor, *Some remarks on graph equation* $G^2 = \overline{G}$, Univ. Beograd. Publ. Elektrotehn. Fak. Ser. Mat., 5 (1994), pp. 43–48.

[Cap79] M. F. Capobianco, *Graph equations*, Ann. New York Acad. Sci., 319 (1979), pp. 114–118.

[CK95] M. Capobianco and S.-R. Kim, *More results on the graph equation* $G^2 = \overline{G}$, in Graph Theory, Combinatorics, and Algorithms, Wiley, New York, 1995, pp. 617–628.

[CLR89] M. F. Capobianco, K. Losi, and B. Riley, $G^2 = \overline{G}$ *has no nontrivial tree solutions*, Ann. New York Acad. Sci., 555 (1989), pp. 103–105.

[CS77] D. M. Cvetković and S. K. Simić, *Graph equations*, in Contributions to Graph Theory and Its Applications, Technische Hochschule Ilmenau, Ilmenau, Germany, 1977, pp. 40–56.

[CS79] D. M. Cvetković and S. K. Simić, *A bibliography of graph equations*, J. Graph Theory, 3 (1979), pp. 311–324.

[Die97] R. Diestel, *Graph Theory*, Grad. Texts in Math. 173, Springer-Verlag, New York, 1997.

[HU79] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, Reading, MA, 1979.

[Sch99] M. Schaefer, *Completeness and Incompleteness*, Ph.D. thesis, University of Chicago, Chicago, 1999.

[Sch01] M. Schaefer, *Graph Ramsey theory and the polynomial hierarchy*, J. Comput. System Sci., 62 (2001), pp. 290–322.

SIAM J. Discrete Math.
Vol. 19, No. 3, pp. 744–761

THE PMU PLACEMENT PROBLEM*

DENNIS J. BRUENI† AND LENWOOD S. HEATH‡

Abstract. The PMU placement problem is an optimization problem abstracted from an approach to supervising an electrical power system. The power system is modeled as a graph, and adequate supervision of the system requires that the voltage at each node and the current through each edge be observable. A phasor measurement unit (PMU) is a monitor that can be placed at a node to directly observe the voltage at that node, as well as the current and its phase through all incident edges. The PMU placement problem is to place PMUs at a minimum number of nodes so that the entire electric power system is observed. A new simpler definition of graph observability and several complexity results for the PMU placement problem are presented. The PMU placement problem is shown to be NP-complete even for planar bipartite graphs. Several fundamental properties of PMU placements are proven, including the property that a minimum PMU placement requires no more than 1/3 of the nodes in a connected graph of at least 3 nodes.

Key words. phasor measurement unit, power system graph observability, domination, electric power monitoring, NP-completeness

AMS subject classifications. 05C69, 05C70, 05C85, 68R10, 94C15

DOI. 10.1137/S0895480103432556

1. Power systems and PMUs.

An electrical power system includes a set of *buses* and a set of *transmission lines* connecting the buses. A *bus* is a substation where lines are joined. A power system also includes a set of *generators,* which supply power, and a set of *loads,* into which the power is directed. To securely control a power system, its state must be monitored [8, 14, 19]. The state of a power system is expressed in terms of state variables, such as voltage at a load and phase angle at a generator. Typically, measurement devices are placed at selected points in the power system to monitor values of the state variables, which are fed back to the central control. The central control adjusts the power system to compensate for imbalances and to prevent hazardous (e.g., fault) situations [23]. For proper control, it is essential that all state variables be communicated to the central control in real time.

A phasor measurement unit (PMU) is a measurement device placed on a bus to monitor voltage at the bus and current phase along outgoing lines [5, 11, 12, 13, 25]. The ability to measure the current phasors as well as the voltage gives the PMU an advantage over other measurement units, enabling the deployment of fewer PMUs than is required in other types of measurement systems, some of which require one measurement unit per bus. PMUs track transients in the power system at high sampling rates, allowing automated real-time monitoring and control [21]. It is important to place the PMUs on buses so as to minimize their number while maintaining system observability, as PMUs are expensive [1, 18].

Stability problems of real-time control using PMUs have been studied before, including neural network approaches to control [16, 17]. Synchronization of the control unit and the PMUs may be done by satellite, using the global positioning system [3, 4, 20, 24], and communications of measurements can be implemented via the

*Received by the editors July 31, 2003; accepted for publication (in revised form) March 1, 2005; published electronically December 7, 2005.

http://www.siam.org/journals/sidma/19-3/43255.html

†SlickEdit Inc., 3000 Aerial Center Parkway, Suite 120, Morrisville, NC 27560 (dennis@slickedit.com).

‡Department of Computer Science, Virginia Tech, Blacksburg, VA 24061-0106 (heath@vt.edu).

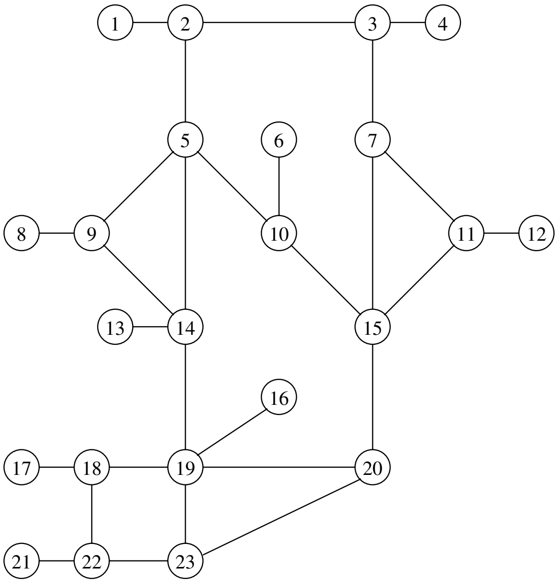


FIG. 1. *Sample power system graph.*

Internet [22]. The problem of optimal placement of PMUs has been studied before. El-Shal and Thorp [6] give an algorithm to optimally place two PMUs to minimize their notion of measurement error. Palmer and Ledwich [20] propose an optimization algorithm based on measurement sensitivity. Baldwin, Mili, Boisen, and Adapa [1] first formulate the PMU placement problem as a problem of minimizing cost and investigate heuristics for the problem.

Brueni [2] recasts the PMU placement problem in a more formal graph-theoretic setting. Haynes, Hedetniemi, Hedetniemi, and Henning [9] also study the problem in a graph-theoretic setting, using the notion of a power dominating set in a graph. Specifically, a power system is modeled as an undirected graph $G = (V, E)$, where V is the set of buses, generators, and loads, and where an edge $(u, v) \in E$ exists if there is a transmission line connecting u and v. For convenience of discussion, such a graph G is called a *power system graph (PSG).* A *PMU placement* Π is a subset of V on which PMUs are placed. System observability is defined as a function of a PSG G and a PMU placement Π that returns the subgraph of G that is observed by Π (see section 2 for the precise definition of observability). A *PMU cover* Π of G is a placement that observes all of G. A *minimum PMU cover* is a PMU cover Π whose size $|\Pi|$ is minimum. Given a PSG G, the PMU placement problem is to find a minimum PMU cover for G. A more formal definition of the problem, together with an example, is given in section 3. Without loss of generality, we assume henceforth that a PSG is a connected graph with at least two nodes.

We make a few observations about a typical PSG, which are illustrated by the sample PSG in Figure 1. A PSG is planar or nearly so; it is uncommon for power lines to intersect, except, of course, at a bus. A PSG has large induced subgraphs that are trees, due to the fact that power distribution is most economical using only a tree; cycles in power systems provide redundancy. A PSG has many degree one nodes—generators and loads. The maximum degree of a PSG is low, because it is impractical to connect a bus to a large number of lines.

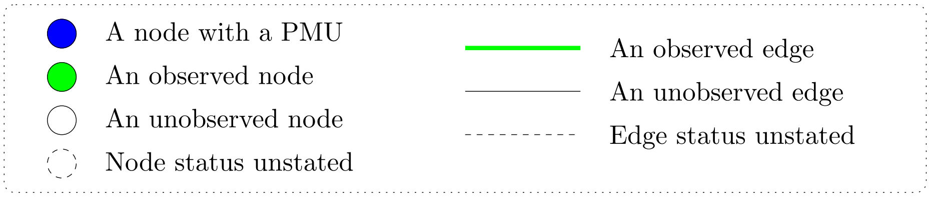


FIG. 2. *Graphical notation for PMU observability.*

In this paper, we address observability and PMU placement as graph-theoretic and algorithmic problems. In section 2, we first take the definition of observability from the power system literature [1] and give an equivalent, much simplified, graph-theoretic definition. Employing the simplified definition of observability, we show how to compute observability in linear time. In section 3, we formally define the minimum PMU placement problem and explore its graph-theoretic properties. In particular, we show that a PSG of at least 3 nodes requires a PMU cover that occupies no more than 1/3 of its nodes. Finally, in section 4, we prove that the PMU placement problem is NP-complete even for planar bipartite graphs.

2. Observability.

2.1. Definitions of observability.

In this section, we provide two definitions of PSG observability and prove the two definitions equivalent. We require some notation and terminology. Fix a PSG $G = (V, E)$. Let $V' \subseteq V$. The *node induced subgraph* $<V'>$ of an undirected graph $G = (V, E)$ is

$$<V'>= (V', \{(u, v) \mid (u, v) \in E \text{ and } u, v \in V'\}),$$

where $V' \subseteq V$. For any node v, its *(open) neighborhood* is $\Gamma_G(v) = \Gamma(v) = \{u \in V \mid (u, v) \in E\}$. Its *closed neighborhood* is $\Gamma_G[v] = \Gamma[v] = \Gamma(v) \cup \{v\}$. A *placement* $\Pi \subseteq V$ is a set of the buses on which PMUs are placed. A bus or a line is *observed* if its state variables are monitored. A PMU *cover* Π is a placement where the entire graph is observed. Figure 2 summarizes the graphical notation used for observability in the remainder of the paper.

Baldwin, Mili, Boisen, and Adapa [1] develop the rules in the following definition of the nodes and edges observed. The rules follow from elementary laws of electrical networks.

DEFINITION 1 (Observability). *Let Π be a placement of PMUs on the nodes of $G = (V, E)$. These rules determine the set of observed nodes Π^R and the set of observed edges Π^-.*

R1. *By definition: A bus with a PMU and any line extending from the bus is observed. Formally, if $v \in \Pi$ and $u \in \Gamma(v)$, then $v \in \Pi^R$ and $(v, u) \in \Pi^-$.*

R2. *Ohm's law, $P = IR$: Any bus that is incident to an observed line connected to an observed bus is observed (the known current in the line, the known voltage at the observed bus, and the known resistance of the line determines the voltage at the bus). Formally, if $(u, v) \in \Pi^-$ and $u \in \Pi^R$, then $v \in \Pi^R$.*

R3. *Ohm's law, $I = P/R$: Any line joining two observed buses is observed (the known voltage at both observed buses and the known resistance of the line determines the current on the line). Formally, if $u, v \in \Pi^R$ and $(u, v) \in E$, then $(u, v) \in \Pi^-$.*

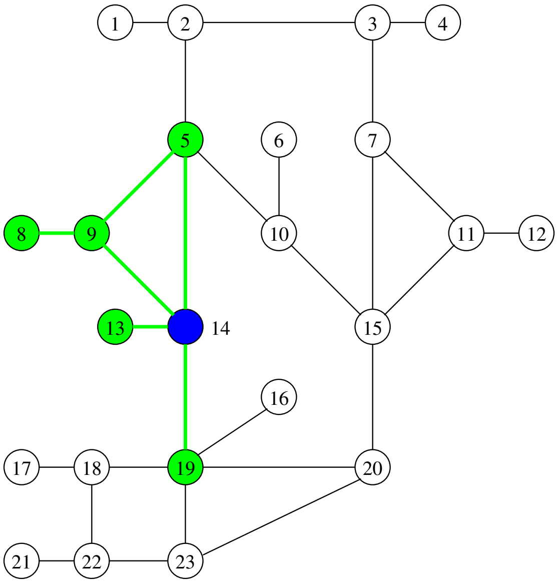


Fig. 3. *Example of definition of observability.*

R4. *Kirchoff current: If all the lines incident to an observed bus are observed, save one, then all of the lines incident to that bus are observed (the net current flowing through a bus is zero). Formally, if $v \in \Pi^R$ and $|\Gamma(v) \cap (V - \Pi^R)| \leq 1$, then $\Gamma[v] \subseteq \Pi^R$.*

R5. *Derived: Any bus incident only to observed lines is observed. Formally, if, for all $u \in \Gamma(v)$, $(v, u) \in \Pi^-$, then $v \in \Pi^R$.*

*Proof. An observed line must be connected to at least one observed bus (*R1 *and* R3*). If all lines incident to a bus are observed, the bus must either be observed itself or each bus adjacent to it is observed. Hence, by* R2*, the bus is observed.* □

This definition does not take into account any inductance or capacitance in the system, which will have effects on the dynamic behavior of the system.

To illustrate the definition, consider the graph of Figure 1 and the placement $\Pi = \{14\}$. Since $14 \in \Pi$, by rule R1, we have

$$14 \in \Pi^R$$
$$(5, 14), (9, 14), (13, 14), (19, 14) \in \Pi^-.$$

By R3, we have $(5, 9) \in \Pi^-$, as $5, 9 \in \Pi^R$. By R4, we have $(8, 9) \in \Pi^-$, as 2 of the 3 lines incident to bus 9 are known to be observed. Finally, we have $8 \in \Pi^R$ by R2; see Figure 3 for the annotated result.

We now provide a simplified definition of observability (originally in Brueni [2]) that requires only 2 rules. Our definition of observability is restricted to observing nodes (buses).

Definition 2 (Simplified Observability). *Let Π be a placement of PMUs on the nodes of $G = (V, E)$. The two rules below determine the set of observed nodes $\Pi^S \subseteq V$.*

S1. *If a node v has a PMU, then all nodes in $\Gamma[v]$ are observed. Formally, if $v \in \Pi$, then $\Gamma[v] \subseteq \Pi^S$.*

S2. *If a node v is observed and all nodes in $\Gamma(v)$ are observed, save one, then all nodes in $\Gamma[v]$ are observed. Formally, if $v \in \Pi^S$ and $|\Gamma(v) \cap (V - \Pi^S)| \leq 1$, then $\Gamma[v] \subseteq \Pi^S$.*

We now demonstrate that Definitions 1 and 2 are equivalent.

THEOREM 1. *Let $G = (V, E)$ be a PSG, and let $\Pi \subseteq V$ be a placement. Then $\Pi^R = \Pi^S$.*

Proof. We first show that $\Pi^S \subset \Pi^R$. The set Π^S can be obtained one node at a time by a sequence of applications of S1 and S2. For purposes of induction, choose a sequence of steps—applications of S1 and S2—that yields Π^S. The base case of the induction is zero steps, in which case the set of nodes obtained is $\emptyset \subset \Pi^R$. Now assume that $v \in \Pi^S$ is obtained in step $k > 0$ and that all nodes obtained at earlier steps are in Π^R. If step k is an S1 step, then either $v \in \Pi$ and $v \in \Pi^R$ by R1 or there is a node $u \in \Gamma(v) \cap \Pi$, in which case $(u, v) \in \Pi^-$ by R1 and $v \in \Pi^R$ by R2. If step k is an S2 step, then there exists an observed node u such that $v \in \Gamma(u)$ and every node $w \in (V - \Pi^S)$ is obtained in an earlier step and hence is in Π^R. By R4, $v \in \Pi^R$. By induction, we conclude that $\Pi^S \subset \Pi^R$.

We now show that $\Pi^R \subset \Pi^S$. The set Π^R can be obtained one node at a time by a sequence of applications of R1–R4 (R5 is derived and an application of R5 can be rewritten using applications of R1–R4). For purposes of induction, choose a sequence of steps—applications of R1–R4—that yields Π^R. The base case of the induction is zero steps, in which case the set of nodes obtained is $\emptyset \subset \Pi^S$. Now assume that $v \in \Pi^R$ is obtained in step $k > 0$ and that all nodes obtained at earlier steps are in Π^S. If step k is an R1 step, then $v \in \Pi$ and $v \in \Pi^S$ by S1. If step k is an R2 step, then there exists an observed node u such that $v \in \Gamma(u)$, $(u, v) \in \Pi^-$, and $u \in \Pi^R$. If $u \in \Pi$, then $v \in \Pi^S$ by S1. Otherwise, $(u, v) \in \Pi^-$ because of an R3 step, at which point $u, v \in \Pi^R$ and hence $v \in \Pi^S$. Rule R3 only observes edges, so an R3 step does not place any node in Π^R. If step k is an R4 step, then there exists an observed node u such that $v \in \Gamma(u)$ and every node $w \in (V - \Pi^R)$ is obtained in an earlier step and hence is in Π^S. By S2, $v \in \Pi^S$. By induction, we conclude that $\Pi^R \subset \Pi^S$.

The theorem follows. □

By eliminating the concern for observing edges, this definition simplifies proofs and algorithms. All results in this paper are presented using Definition 2.

2.2. Observability computation in linear time. The computation of Π^S for a PSG $G = (V, E)$ can be accomplished in time linear in $|V| + |E|$; see Algorithm OBSERVE in Figure 4. (The algorithm of Haynes, Hedetniemi, Hedetniemi, and Henning [9] that implements Definition 1 is not obviously linear time.) For each node $v \in V$, the variable *observedneighbors*$[v]$ maintains the number of nodes in $\Gamma(v)$ that are currently known to be observed. The *degree* of v is $\text{DEGREE}(v) = |\Gamma(v)|$.

THEOREM 2. *For $G = (V, E)$ and $\Pi \subseteq V$, Algorithm* OBSERVE *computes Π^S in $O(|V| + |E|)$ time.*

Proof. An examination of Algorithm OBSERVE shows that it implements rules S1 and S2 of Definition 2. The **for** loop for rule S1 marks all the nodes in Π and all their neighbors observed. To implement rule S2, every node u that is observed and whose observed neighbor count reaches the S2 threshhold of $\text{DEGREE}(u) - 1$ is placed in the queue Q. In the rule S1 **for** loop, neighbors of nodes in Π that reach the S2 threshhold are enqueued. (There is no need to enqueue a node whose observed neighbor count equals its degree.) The **while** loop implements the propagation of observation of rule S2. Each dequeued node v was enqueued at a time when it was already marked observed and had observed neighbor count $\text{DEGREE}(v) - 1$. At the time it is dequeued,

```
Observe($G, \Pi$)
$Q \leftarrow \emptyset$
for each $v \in V$
    do observed[$v$] $\leftarrow$ false
for each $v \in V$
    do observedneighbors[$v$] $\leftarrow 0$
for each $v \in \Pi$
    do $\triangleright$ Rule S1—observe all elements of $\Pi$ and their neighbors
        for each $u \in \Gamma[v]$
            do if not observed[$u$]
                    then  observed[$u$] $\leftarrow$ true
                          for each $w \in \Gamma(u)$
                              do observedneighbors[$w$] $\leftarrow$ observedneighbors[$w$] $+ 1$
        for each $u \in \Gamma(v)$
                do $\triangleright$ Enqueue neighbors of $\Pi$ that reach the S2 threshold
                    if observedneighbors[$u$] $=$ DEGREE($u$) $- 1$
                        then  Enqueue($Q, u$)
while $Q \neq \emptyset$
    do $\triangleright$ $v$ is observed and has at most one unobserved neighbor
        $v \leftarrow$ Dequeue($Q, w$)
        if observedneighbors[$v$] $=$ DEGREE($v$) $- 1$
            then  $u \leftarrow$ unobserved neighbor of $v$
                    observed[$u$] $\leftarrow$ true
                    for each $w \in \Gamma(u)$
                        do observedneighbors[$w$] $\leftarrow$ observedneighbors[$w$] $+ 1$
                            if observed($w$)
                                if observedneighbors[$w$] $=$ DEGREE($w$) $- 1$
                                    then  Enqueue($Q, w$)
                    if observedneighbors[$u$] $=$ DEGREE($u$) $- 1$
                        then  Enqueue($Q, u$)
return $\{v \in V \mid$ observed[$v$]$\}$
```

FIG. 4. *Algorithm Observe to compute the observability function.*

the observed neighbor count of v may have increased to DEGREE(v). Otherwise, v has a unique neighbor u that is not marked observed. It is u that becomes observed as a consequence of rule S2. In both places in OBSERVE where a node u is marked observed, the count of observed neighbors of u is incremented, so that the *observedneighbors* values are correctly maintained. Moreover, in the **while** loop, whenever an observed node reaches the S2 threshhold, it is enqueued. We conclude that Algorithm OBSERVE correctly computes Π^S.

Every node is marked observed at most once and is enqueued at most once. The tests for the S2 threshhold are executed at most $|E|$ times and require at most $(|E|)$ work. The remaining work is done at most once for each node and is hence $O(|V|)$. We conclude that the time complexity of OBSERVE is $O(|V| + |E|)$. □

3. Properties of PMU placement. In this section, we explore graph-theoretic properties of PMU placement.

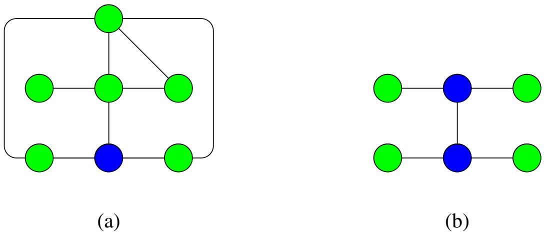


FIG. 5. *Minimum covers for* (a) *a graph G and* (b) *an induced subgraph of G.*

3.1. The PMU placement problem. The PMU placement problem (PMUP) of finding a minimum cover is stated formally here.

Problem: PMU Placement (Optimization Version)

INSTANCE: Graph $G = (V, E)$.

QUESTION: Find a cover $\Pi \subseteq V$ such that for any cover $\Pi' \subseteq V$, $|\Pi| \leq |\Pi'|$.

Such a placement Π is called a *minimum PMU cover*. The reader may verify, with some effort, that $\Pi = \{3, 10, 14, 19, 22\}$ is a minimum PMU cover for the PSG of Figure 1.

Haynes, Hedetniemi, Hedetniemi, and Henning [9] call the same problem the power domination problem (PDS) and explore the analogy between PDS and the traditional domination set problem. Though both problems involve some kind of observation of part of a graph, there is the significant difference that observation in dominating sets has bounded locality, while observation in PMUP can propagate more globally. For example, a single PMU suffices to observe a path or cycle PSG. Given an undirected graph G, a dominating set for G is also a PMU cover for G, although it is a poor one in many cases. The converse is, of course, seldom true.

3.2. Induced subgraphs. One might expect that an induced subgraph of a PSG G would always have a minimum PMU cover no larger than the size of a minimum PMU cover of G. However, this expectation is incorrect, as illustrated by the graph G in Figure 5. The single PMU in Figure 5(a) directly observes three nodes, two of which are of degree two. These degree two nodes then allow the node at the top to be observed, after which the observability of the remaining two nodes follows. While the graph in Figure 5(b) is induced by all but one of the nodes of G, it is clearly impossible to observe all of this subgraph of G without two PMUs.

3.3. Placement substitution. The following theorem shows that certain placement sets may replace others.

THEOREM 3 (Substitution). *Given a PSG $G = (V, E)$ and two placements $\Pi_1, \Pi_2 \subseteq V$, if ${\Pi_1}^S \subseteq {\Pi_2}^S$, then for any placement Π, $(\Pi \cup \Pi_1)^S \subseteq (\Pi \cup \Pi_2)^S$.*

Proof. For purposes of induction, choose a sequence of steps—applications of S1 and S2—that yields $(\Pi \cup \Pi_1)^S$. The base case of the induction is zero steps, in which case the set of nodes obtained is $\emptyset \subset (\Pi \cup \Pi_2)^S$. Now assume that $v \in (\Pi \cup \Pi_1)^S$ is obtained in step $k > 0$ and that all nodes obtained at earlier steps are in $(\Pi \cup \Pi_2)^S$. If step k is an S1 step, then there exists $u \in \Pi \cup \Pi_1$ such that $v \in \Gamma[u]$. If $u \in \Pi$, then $v \in (\Pi \cup \Pi_2)^S$ by S1. If $u \in \Pi_1$, then $v \in (\Pi \cup \Pi_2)^S$ since ${\Pi_1}^S \subseteq {\Pi_2}^S$. If step k is an S2 step, then there exists an observed node u such that $v \in \Gamma(u)$ and every node $w \in (V - (\Pi \cup \Pi_1)^S)$ is obtained in an earlier step and hence is in $(\Pi \cup \Pi_2)^S$. By S2, $v \in (\Pi \cup \Pi_2)^S$. By induction, we conclude that $(\Pi \cup \Pi_2)^S \subset (\Pi \cup \Pi_2)^S$. □

If $|\Pi_2| < |\Pi_1|$ with ${\Pi_1}^S \subseteq {\Pi_2}^S$, then substituting Π_2 for Π_1 in a PMU cover results in a smaller cover, without loss of system observability.

The following corollary to Theorem 3 shows that it is counterproductive to place a PMU on a degree one node (unless, of course, $|V| = 2$).

Corollary 1. *Given a PSG $G = (V, E)$ with a cover $\Pi \subseteq V$ such that there is a degree one node $v \in \Pi$, there exists a cover Π' such that $v \notin \Pi'$ and $|\Pi'| \leq |\Pi|$.*

Proof. Let $\{u\} = \Gamma(v)$ and $\Pi' = (\Pi - \{v\}) \cup \{u\}$. Clearly, $\{v\}^S \subseteq \{u\}^S$. By Theorem 3, Π' is a PMU cover for G such that $v \notin \Pi'$ and $|\Pi'| \leq |\Pi|$. □

A second corollary shows that it is counterproductive to place a PMU on a degree two node (unless, of course, G is a path or a cycle).

Corollary 2. *Given a PSG $G = (V, E)$ with a cover $\Pi \subseteq V$ such that there is a degree two node $v \in \Pi$, there exists a cover Π' such that $v \notin \Pi'$ and $|\Pi'| \leq |\Pi|$.*

Proof. Let $\{u, w\} = \Gamma(v)$ and $\Pi' = (\Pi - \{v\}) \cup \{u\}$. Note that $w \in \{u\}^S$ by application of S1 and S2. Since $\Gamma[v] \subseteq \{u\}^S$, we have $\{v\}^S \subseteq \{u\}^S$. By Theorem 3, Π' is a PMU cover for G such that $v \notin \Pi'$ and $|\Pi'| \leq |\Pi|$. □

Corollaries 1 and 2 are implicit in Observation 4 of Haynes, Hedetniemi, Hedetniemi, and Henning [9].

3.4. Placing a PMU on a separation node.

A *separation node* in a connected graph is one whose removal leaves a subgraph with two or more components. Baldwin, Mili, Boisen, and Adapa [1] claim that if a PMU placed at a separation node v observes all of the nodes in any one of the subgraphs resulting from the deletion of v, then v is an element of some minimum cover. This claim may fail if the observed subgraph is a path, due to the propagation of observability using S2, even when v has no PMU. The following restatement is correct.

Theorem 4. *Let $G = (V, E)$ have separation node x. Let $u, w \in \Gamma(x)$ be distinct nodes. Let U and W be the components of $<V - x>$ containing u and w, respectively. If $U \cup W \subseteq \{x\}^S$, then there exists a minimum cover for G containing x.*

Proof. Note that U and W do not have to be distinct. Let Π_1 be any minimum PMU cover of G. If $x \in \Pi_1$, then we are done. Otherwise, by S2, there must be a node $y \in (U \cup W) \cap \Pi_1$. Let $\Pi_2 = \{x\} \cup (\Pi_1 - \{y\})$. Then Π_2 is a minimum cover for G containing x. □

3.5. Upper bound on the size of a minimum PMU cover.

In this section, we show that, in a PSG having $n \geq 3$ nodes, at most $\lfloor n/3 \rfloor$ PMUs suffice to cover the PSG and that this upper bound is tight. Haynes, Hedetniemi, Hedetniemi, and Henning [9] show the same upper bound just for trees (their Theorem 14).

In a PSG, a node u is *symmetric* to a node v, written $u \equiv v$, if $\Gamma(u) - \{v\} = \Gamma(v) - \{u\}$.

Theorem 5. *Node symmetry is an equivalence relation.*

Proof. Let $G = (V, E)$ be a connected graph. *Reflexivity.* For any $x \in V$, $\Gamma(x) - \{x\} = \Gamma(x) - \{x\}$ and hence $x \equiv x$. *Symmetry.* For any $x, y \in V$, $x \equiv y$ implies $\Gamma(x) - \{y\} = \Gamma(y) - \{x\}$, which implies $y \equiv x$. *Transitivity.* For any $x, y, z \in V$, $x \equiv y$ and $y \equiv z$ implies $\Gamma(x) - \{y\} = \Gamma(y) - \{x\}$ and $\Gamma(y) - \{z\} = \Gamma(z) - \{y\}$. These imply that $(x, z) \in E$ if and only if $(y, z) \in E$ and $(x, y) \in E$ if and only if $(x, z) \in E$. Consequently, $(x, y) \in E$ if and only if $(y, z) \in E$. Let $N = \Gamma(x) \cup \Gamma(y) \cup \Gamma(z) - \{x, y, z\}$. Thus $\Gamma(x) - \{z\} = (\Gamma(x) \cap \{y\}) \cup N = (\Gamma(z) \cap \{y\}) \cup N = \Gamma(z) - \{x\}$. Hence, $x \equiv z$.

Thus, node symmetry is an equivalence relation. □

For a PSG $G = (V, E)$, let S be the set of equivalence classes of V under $\equiv$. For every $P \in S$, $<P>$ is either a clique or an independent set. For two distinct equivalence classes $P, Q \in S$, P is *adjacent* to Q if for every $u \in P$, we have $Q \subseteq \Gamma(u)$. Note that P adjacent to Q implies Q adjacent to P. Define $A(S) = \{(P, Q) \mid P, Q \in S \text{ and } P \text{ adjacent to } Q\}$. The graph $H(S) = (S, A(S))$ is the *adjacency graph* of S. For any $R \subseteq S$, define $\pi(R) = \cup_{U \in R} U$.

LEMMA 1. *Let $G = (V, E)$ be a PSG, and let S be the set of equivalence classes of V under $\equiv$. Let $U_1, U_2 \in V$ be distinct equivalence classes, and let $u_1 \in U_1$ and $u_2 \in U_2$. Then $(u_1, u_2) \in E$ if and only if $(U_1, U_2) \in A(S)$. Consequently, $H(S)$ is connected.*

An equivalence class of $\equiv$ containing more than one node represents a kind of node redundancy. The following lemma identifies a small placement that dominates all but one node of each equivalence class.

LEMMA 2. *Let $G = (V, E)$ have 3 or more nodes. There exists a placement Π such that*

1. *for every distinct $u, v \in V$ such that $u \equiv v$, either $u \in \Gamma[\Pi]$ or $v \in \Gamma[\Pi]$; and for every $U \in S$, $|\Gamma[\Pi] \cap U| \geq |U| - 1$; and*
2. $|\Gamma[\Pi]| \geq 3|\Pi|$.

Proof. First suppose that G is a clique. Then $S = \{V\}$ and $H(S) = (S, \emptyset)$. Let $\Pi = \{v\}$, where $v \in V$. Clearly, Π satisfies (1) and (2).

Now suppose that G is not a clique. Then $|S| \geq 2$. We proceed by induction on $|S|$ to show that there exists a Π that satisfies (1) and (2), as long as $|\pi(S)| \geq 3$. The base case is $|S| = 2$. Let $S = \{U_1, U_2\}$, where $|U_1| \geq |U_2| \geq 1$. If $|U_2| = 1$ or $|U_2| = 2$, then let $\Pi = \{u\}$ for any $u \in U_2$. If $|U_2| \geq 3$, then let $\Pi = \{u_1, u_2\}$ for any $u_1 \in U_1$ and any $u_2 \in U_2$. In both cases, $U_1 \cup U_2 \subseteq \Gamma[\Pi]$, and $|U_1 \cup U_2| \geq 3|\Pi|$. Hence, (1) and (2) hold for Π.

Now assume that $|S| = m \geq 3$ and that the inductive hypothesis holds for any adjacency graph $H(S')$ of size less than m, as long as $|\pi(S')| \geq 3$. Let $T = (S, F)$ be a spanning tree of $H(S)$. Choose $U \in S$ that is not a leaf but is adjacent to at least one leaf in T. Root T at U. Let $T_1, T_2, \ldots, T_r$ be the subtrees under U. Note that $r \geq 2$, since $|S| \geq 3$. For $1 \leq j \leq r$, let R_j be the root of T_j. Without loss of generality, assume that R_j is a leaf of T for $j \leq s$ and a nonleaf for $j > s$, where $1 \leq s \leq r$, and that the T_j, for $1 \leq j \leq s$, are arranged in nondecreasing order by cardinality of $|R_j|$.

First suppose that $|R_1| = 1$. Then R_1 places no constraints on Π with respect to (1) or (2). Let $S' = S - \{R_1\}$. If $|S'| \geq 3$, then, by induction, a placement Π can be found for $H(S')$ that satisfies (1) and (2) for $<V - R_1>$ and hence also for G. If $|S'| = 2$, then $s = r = 2$. Select $u \in U$ and $w \in R_2$. If $|R_2| \leq 2$, then set $\Pi = \{w\}$. If $|R_2| \geq 3$ and $|U| \leq 2$, then set $\Pi = \{u\}$. If $|R_2| \geq 3$ and $|U| \geq 3$, then set $\Pi = \{u, w\}$ (in this case, $|\pi(S)| \geq 6$). In all cases, Π satisfies (1) and (2) for G.

Now suppose that $|R_1| \geq 2$. Select $u \in U$. Consider the case $|U| \leq 2$. If $r = s$, then set $\Pi = \{u\}$. Otherwise, consider each T_j, where $s + 1 \leq j \leq r$, in turn. If $|\pi(T_j)| \geq 3$, then apply the inductive hypothesis to T_j to identify Π_j that satisfies (1) and (2) for T_j. If $|\pi(T_j)| \leq 2$, then T_j is a path of one-node equivalence classes; set $\Pi_j = \emptyset$. Set $\Pi = \{u\} \cup \bigcup_{j=s+1}^{r} \Pi_j$. Then Π satisfies (1) and (2). Now consider the case $|U| \geq 3$. In this case, $|U - \{u\}| \geq 2$ and $| <V - R_1 - \{u\}> | \geq 3$. Let $G' = <V - R_1 - \{u\}>$. By induction, there exists a Π' satisfying (1) and (2) for G'. Set $\Pi = \Pi' \cup \{u\}$. Then Π satisfies (1) and (2).

By induction, we obtain $\Pi \subseteq V$ satisfying (1) and (2). □

THEOREM 6. *Let $G = (V, E)$ be a PSG, and let $n = |V|$. Then there exists a cover Π satisfying $|\Pi| \leq \lfloor n/3 \rfloor$, if $n \geq 3$, and $|\Pi| = 1$, if $1 \leq n \leq 2$.*

Proof. The result for $1 \leq n \leq 2$ is immediate. For $n \geq 3$, the proof is an inductive

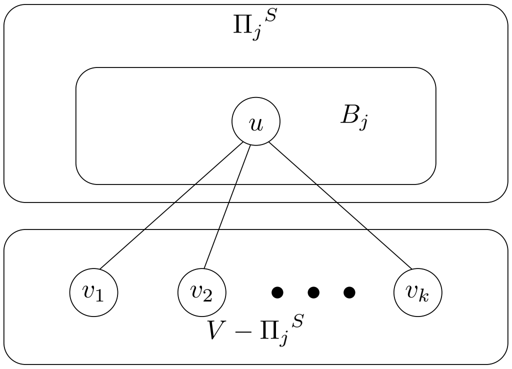


FIG. 6. *Boundary node $u \in B$.*

construction of a sequence of placements $\Pi_0{}^S, \Pi_1{}^S, \dots, \Pi_\ell{}^S$ such that, for $0 \le j < \ell$, we have that Π_j is a proper subset of Π_{j+1}; Π_ℓ is a cover of G; and, for $0 \le j \le \ell$, we have $\Pi_j \neq \emptyset$ and $|\Pi_j{}^S \ge 3|\Pi_j{}^S|$.

The base case is $j = 0$. Let Π' be the initial placement guaranteed by Lemma 2. If $\Pi' \neq \emptyset$, then set $\Pi_0 = \Pi'$. Otherwise, set $\Pi_0 = \{u\}$, where u is any degree 2 node of G. Clearly, $\Pi_0 \neq \emptyset$ and $|\Pi_j{}^S \ge 3|\Pi_j{}^S|$, as required.

Now suppose that $j \ge 0$ and that, for every $0 \le i < j$, Π_i is a proper subset of Π_{i+1}, and, for $0 \le i \le j$, $\Pi_j \neq \emptyset$ and $|\Pi_i{}^S \ge 3|\Pi_i{}^S|$. Let $B_j = \{u \in \Pi_j{}^S \mid \Gamma(u) \cap (V - \Pi_j{}^S)\}$ be the set of *boundary nodes*—observed nodes adjacent to unobserved nodes. If $B_j = \emptyset$, then Π_j is a cover of G and the theorem is proved for G. Otherwise, $V - \Pi_j{}^S \neq \emptyset$. In that case, we construct Π_{j+1} as follows.

Clearly $B_j \cap \Pi_j = \emptyset$, since $\Gamma[\Pi_j] \subset \Pi^S$. Let $u \in B_j$, and let $\Gamma(u) \cap (V - \Pi_j{}^S) = \{v_1, v_2, \dots, v_k\}$, as illustrated in Figure 6. Without loss of generality, we may assume that u is selected so that k is as large as possible. Observe that $k \ge 2$, because if v_1 were the only unobserved neighbor of u, then v_1 would be observed by rule S2.

First consider the case $k \ge 3$. Set $\Pi_{j+1} = \Pi_j \cup \{u\}$, a proper superset of Π_j. Then $|\Pi_{j+1}{}^S| \ge |\Pi_j{}^S| + 3$, as desired.

Now consider the case $k = 2$, which means that every node in B_j is adjacent to exactly two nodes of $V - \Pi_j{}^S$. Without loss of generality, assume that $\text{DEGREE}(v_1) \ge \text{DEGREE}(v_2) \ge 1$. Since $v_1 \not\equiv v_2$, we cannot have $\text{DEGREE}(v_1) = \text{DEGREE}(v_2) = 1$. Thus, $\text{DEGREE}(v_1) \ge 2$.

Let $C_1 = (V_1, E_1)$ (respectively, $C_2 = (V_2, E_2)$) be the component of $<V - \Pi_j{}^S>$ containing v_1 (respectively, v_2). First consider the cases where $|V_1| \ge 3$ or where $|V_1| = 2$ and $C_1 \neq C_2$. Select a $v_3 \in V_1 \cap \Gamma(v_1)$ that is not v_2. Set $\Pi_{j+1} = \Pi_j \cup \{v_1\}$, a proper superset of Π_j. We obtain $v_2, v_3 \in \Pi_{j+1}{}^S$; in particular, $v_2 \in \Pi_{j+1}{}^S$ because v_2 is the last unobserved neighbor of u and hence is observed by rule S2. Therefore, $|\Pi_{j+1}{}^S| \ge |\Pi_j{}^S| + 3$, as desired. Now consider the cases where $|V_1| = 1$ or where $|V_1| = 2$ and $C_1 = C_2$. These cases imply that v_1 and v_2 are adjacent only to nodes in B_j and perhaps each other. Since $\text{DEGREE}(v_1) \ge 2$ and $v_1 \not\equiv v_2$, there must be a node $w \in B_j - \{u\}$ adjacent to v_1 and not adjacent to v_2. Let $\Gamma(w) \cap (V - \Pi_j{}^S) = \{v_1, z\}$; see Figure 7. We have $z \neq v_2$, since w is not adjacent to v_2. Set $\Pi_{j+1} = \Pi_j \cup \{v_1\}$, a proper superset of Π_j. We obtain $v_2, z \in \Pi_{j+1}{}^S$ by application of rule S2 to u and w. Therefore, $|\Pi_{j+1}{}^S| \ge |\Pi_j{}^S| + 3$, as desired.

Since the sequence of placements are increasing, we must eventually reach the

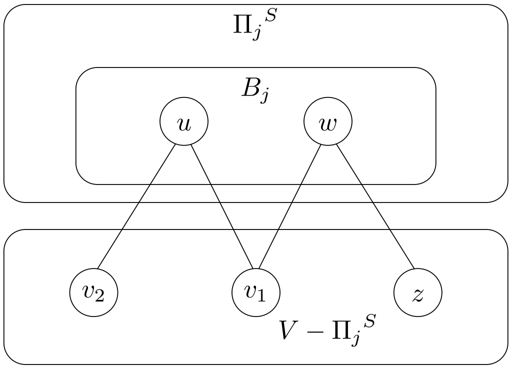


FIG. 7. *Boundary nodes $u, w \in B_j$.*

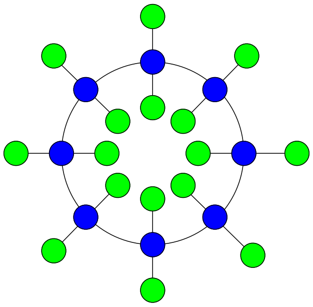


FIG. 8. *Corona $B_{\ell,2}$, which requires $n/3$ PMUs.*

case where $B_j = \emptyset$. The theorem follows. $\square$

We now show that the above bound is existentially tight. To do so, we start with a construction defined in Haynes, Hedetniemi, Hedetniemi, and Henning [9]. If G and H are two graphs, then the *corona* $G \circ H$ of G and H is achieved by making a copy H_v of H for every node v of G and adding an edge from v to every node of H_v. For purposes of notation, let $C_\ell = (U_\ell, E_\ell)$, where

$$\begin{aligned} U_\ell &= \{u_1, u_2, \ldots, u_\ell\} \\ E_\ell &= \bigcup_{i=1}^{\ell-1} \{(u_i, u_{i+1})\} \cup \{(u_\ell, u_1)\}), \end{aligned}$$

be a cycle of length ℓ, and let $I_k = (V_k, \emptyset)$, where $V_k = \{v_1, v_2, \ldots, v_k\}$, be a graph of k isolated nodes. For each $u_i \in U_\ell$, define a copy I_{k,u_i} of I_k by $I_{u_i,k} = (V_{k,u_i}, \emptyset)$, where $V_{u_i,k} = \{v_{i,1}, v_{i,2}, \ldots, v_{i,k}\}$. The corona $B_{\ell,k} = C_\ell \circ I_k$ is a graph of $n = k\ell$ nodes that requires exactly ℓ PMUs to be observed when $k \geq 2$. Moreover, the initial placement phase in the proof of Theorem 6 finds exactly the minimum PMU cover of $B_{\ell,k}$. More specifically, $B_{\ell,2}$ requires exactly $n/3$ PMUs to be observed, which we show in Theorem 7. For example, a minimum PMU cover for $B_{\ell,2}$ has exactly ℓ PMUs as shown in Figure 8.

THEOREM 7. *A minimum cover for $B_{\ell,k}$ requires $\lceil \ell/3 \rceil$ PMUs if $k = 1$ and requires ℓ PMUs if $k \geq 2$.*

Proof. First consider the construction of a minimum PMU cover of $B_{\ell,1} = C_\ell \circ I_1$. Starting with an arbitrary point on C_ℓ, place a PMU on every third node of C_ℓ. It is easy to verify that such a placement is a minimum cover, since every degree one node is either adjacent to a PMU or adjacent to a node of C_ℓ that is adjacent to PMU.

Now assume $k \geq 2$ and consider the construction of a minimum PMU cover of $B_{\ell,k}$. Let $V = \cup_{i=1}^{\ell} V_{u_i,k}$. Let Π be a minimum cover of G. Among all minimum covers of G, select $\Pi \cap V$ to be as small as possible. Suppose $v = v_{i,j} \in \Pi \cap V$. Then v is a degree one node adjacent only to u_i, a degree $k+2$ node. The set $\{u_i\} \cup (\Pi - \{v\})$ is a cover of G of the same cardinality as Π, but with one fewer element of V, contradicting the choice of Π. Hence, $\Pi \cap V = \emptyset$. We claim that $\Pi = U_\ell$. Consider any $u_i \in U_\ell$. We know that none of the k neighbors of u_i in V are in Π. If $u_i \notin \Pi$, then the k neighbors are observed via applications of rule S2. But rule S2 can only be applied when at most one neighbor of u_i is unobserved, while $k \geq 2$. We conclude that $u_i \in \Pi$ and, moreover, that $\Pi = U_\ell$. The theorem follows. □

One referee suggested this generalization of Theorem 7.

THEOREM 8. *Let G be a connected graph with ℓ nodes, and let $k \geq 2$. Then a minimum cover for the corona $G \circ I_k$ requires ℓ PMUs.*

The proof is similar to that of Theorem 7.

4. NP-completeness.

4. NP-completeness. Haynes, Hedetniemi, Hedetniemi, and Henning [9] show that PMUP is NP-complete for bipartite graphs and for chordal graphs. Here we show that the following decision problem version of PMUP is NP-complete even for planar bipartite graphs.

Problem: PMU Placement (Decision Version)

INSTANCE: Graph $G = (V, E)$, integer $k \geq 1$.

QUESTION: Is there a set $\Pi \subseteq V$ such that $|\Pi| \leq k$ and $\Pi^S = V$?

THEOREM 9. *PMUP is NP-complete even when restricted to the class of planar bipartite graphs.*

Proof. The decision problem is easily in NP. Nondeterministically select k nodes forming a candidate Π and verify observability using the methods described in section 2.

The remainder of the proof is a reduction from planar 3-SAT (P3SAT) [15]. An instance of 3-SAT is a boolean formula ϕ in conjunctive normal form such that each clause contains at most 3 literals [7]. ϕ consists of the variables $\{v_1, v_2, \ldots, v_r\}$ and the set of clauses $\{c_1, c_2, \ldots, c_s\}$. Each c_j is a set containing at most 3 literals, where each literal is either a variable v_i or its complement $\overline{v_i}$. A clause containing exactly k literals is called a *k-clause.* The *graph of* ϕ, $G(\phi) = (V(\phi), E(\phi))$, is a bipartite graph constructed as follows:

$$\begin{array}{rcl} V(\phi) & = & \{v_i \mid 1 \leq i \leq r\} \cup \{c_j \mid 1 \leq j \leq s\} \\ E(\phi) & = & \{(v_i, c_j) \mid v_i \in c_j \text{ or } \overline{v_i} \in c_j\}. \end{array}$$

The edges in $E(\phi)$ represent whether a variable occurs in a clause or not. For example, the graph of the formula

$$\phi = (\overline{v_1} \vee v_2 \vee v_3) \wedge (\overline{v_1} \vee \overline{v_4} \vee v_5) \wedge (\overline{v_2} \vee \overline{v_3} \vee \overline{v_5}) \wedge (v_3 \vee \overline{v_4}) \wedge (\overline{v_3} \vee v_4 \vee \overline{v_5})$$

is shown in Figure 9. ϕ is satisfied if $v_2, \overline{v_4}$, and $\overline{v_5}$ are true; hence ϕ is a satisfiable formula. Lichtenstein shows that 3-SAT is NP-complete even when $G(\phi)$ is planar (the problem P3SAT) [15].

It suffices to consider only instances of P3SAT such that each clause contains either 2 or 3 literals. Our planar embedding of $G(\phi)$ positions each node v_i, $1 \leq i \leq r$, along a straight line; this is called the *variable axis.* From Lemma 1 of [15], we may

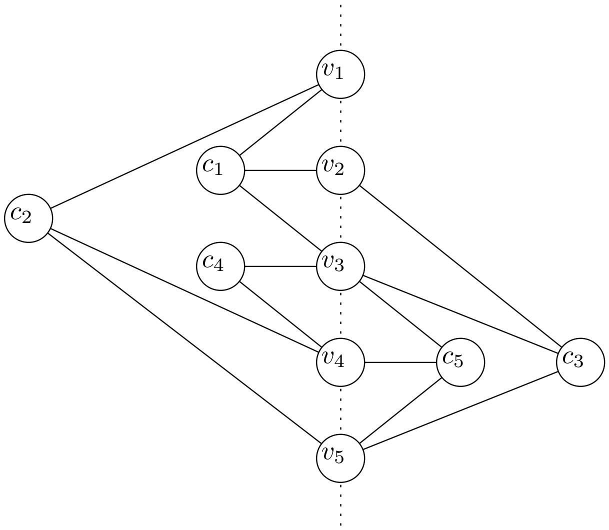


FIG. 9. *Example of planar* 3-*SAT.*

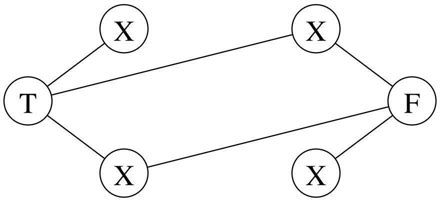


FIG. 10. *A gadget for a variable.*

assume that our planar embedding of $G(\phi)$ satisfies the condition that, for each v_i, all clauses containing the literal v_i are on one side of the variable axis and all clauses containing the literal $\overline{v_i}$ are on the other side. This property of the planar embedding of $G(\phi)$ is called *consistency* [10]. Figure 9 is an example of a consistent planar embedding.

Let $V = \{v_1, \ldots, v_r\}$ and $C = \{c_1, c_2, \ldots, c_s\}$ be an instance of P3SAT such that $G(\phi)$ has a consistent planar embedding. We will construct a corresponding instance of PMUP that also is a planar bipartite graph. The strategy is to replace each node in $G(\phi)$ with a specially constructed graph, or *gadget*. Let $H(\phi)$ denote the resulting graph. Each clause node c_j, $1 \le j \le s$, is replaced with a 2-clique $C[j], C'[j]$, effectively making the clause node adjacent to an additional degree one node. The gadgets placed on clauses simply force a clause to be adjacent to at least one node with a PMU. Each variable node v_i is replaced by the gadget shown in Figure 10. Observe that the gadget forces at least one PMU placed on it in order to be covered. This implies that $H(\phi)$ requires a minimum of r PMUs in its cover. We wish to show that a minimum cover for $H(\phi)$ uses exactly r PMUs if and only if ϕ is satisfiable. Thus we are allowed only one PMU per gadget.

The gadget is designed to toggle between two states, representing either a *true* (T) or *false* (F) value for the literal it replaces, depending on which node the PMU is placed on; see Figure 11. For any variable v_i, let $z_i \in \{v_i, \overline{v_i}\}$ denote the variable

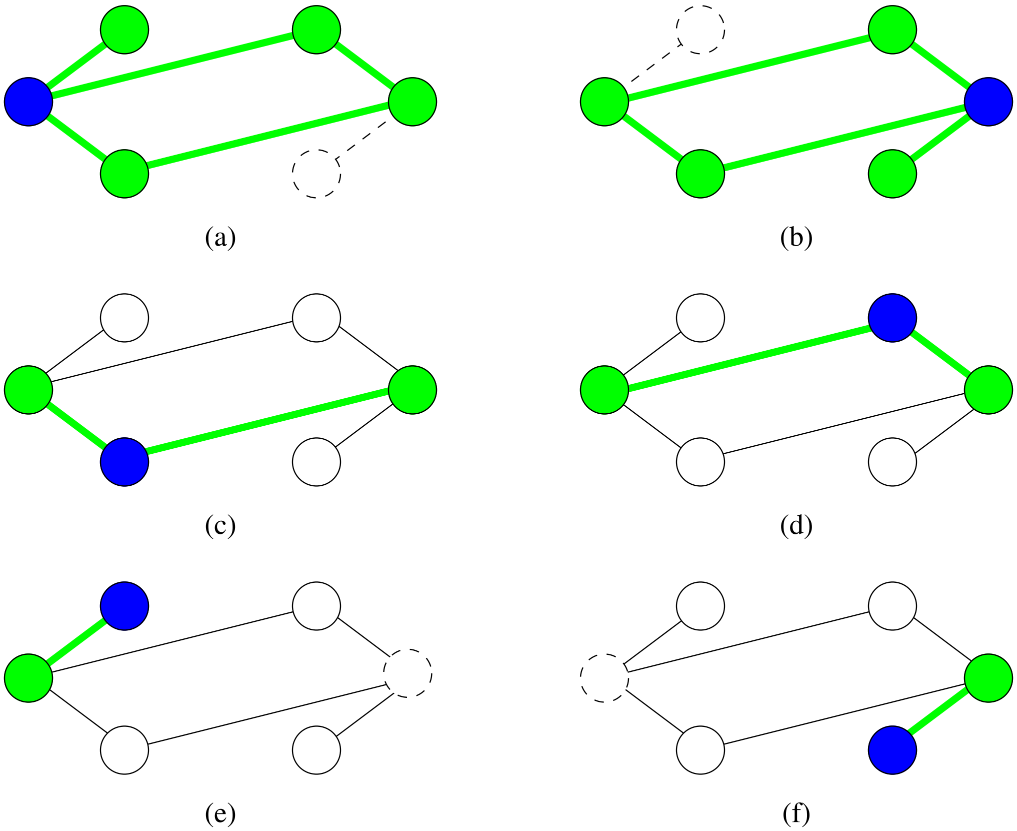


Fig. 11. *Gadget states:* (a) *true;* (b) *false;* (c) *left bridge;* (d) *right bridge;* (e) *left leaf; and* (f) *right leaf.*

appearing in all clauses to the left of the variable axis. The following cases ensue:

1. *true*: In this case, the gadget is indicating that z_i is true. The right leaf of the gadget is observed only if all clauses connected to the rightmost node are observed.
2. *false*: In this case, the gadget is indicating that z_i is false. The left leaf of the gadget is observed only if all clauses connected to the leftmost node are observed.
3. *eliminated (left bridge and right bridge)*: It is impossible to cover the gadget with one PMU on either bridge.
4. *eliminated (left leaf and right leaf)*: It is impossible to cover the gadget with one PMU on a leaf.

For illustration, consider the instance of P3SAT depicted in Figure 9, with gadgets inserted, as shown in Figure 12, and shown with a minimum PMU cover in Figure 13.

We have shown our construction guarantees a graph $H(\phi)$ for which a minimum PMU cover has at least one PMU per gadget. At this time, note that $H(\phi)$ is planar *and* bipartite, as shown in Figure 14. We have also classified the nodes of the gadget semantically as either *true*, *false*, or *illegal*. By the substitution lemma, we do not need to consider illegal nodes when constructing a minimum PMU cover for $H(\phi)$. It remains to show that $H(\phi)$ has a cover of size r if and only if ϕ is satisfiable.

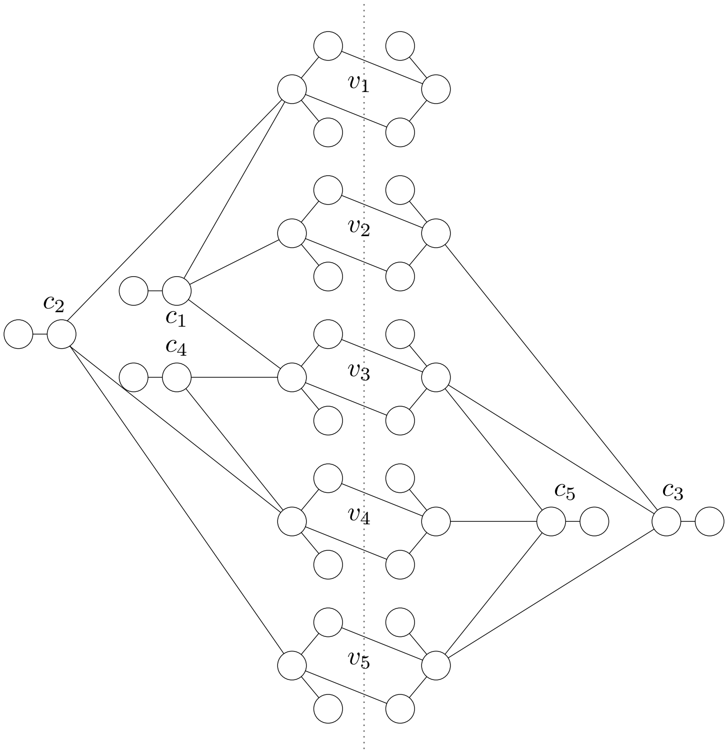


FIG. 12. *Instance of PMUP (planar 3-SAT with gadgets).*

Assume that ϕ is satisfiable. For each variable v_i, $1 \leq i \leq r$, place a PMU on either the leftmost or the rightmost gadget node according to whether v_i is true or false in a given satisfying instance S for ϕ. If ϕ is satisfied, then for each clause c_j, $1 \leq j \leq s$, there exists a literal $v_i \in c_j$ or $\overline{v_i} \in c_j$ which is in S. The PMU placed on the corresponding gadget observes the main node of c_j, as well as the main body of the v_i's gadget. Thus all main clause nodes are observed. Furthermore, all leaf nodes on clauses become observed by S2. Likewise, the remaining leaf nodes on gadgets become observed. Hence, there is a cover of size r for $H(\phi)$.

Now assume that $H(\phi)$ has a cover Π of size r. Each gadget must have at least one PMU. Thus there can be no nongadget PMUs in Π. Since Π is a cover, all clauses are observed. By construction, a clause cannot be observed unless it is adjacent to at least one PMU located on a gadget. Then for each main clause node c_j, $1 \leq j \leq s$, there exists a node $u \in \Gamma(c_j)$ with a PMU. Let v_i be variable containing u. Let $z_i \in \{v_i, \overline{v_i}\}$ be the variable appearing in c_j. The clause c_j is satisfied if z_i is chosen as true. Hence, all clauses in ϕ are satisfied by the truth assignment derived from the minimum cover Π.

In summary, we have transformed instance ϕ of P3SAT into a PSG $H(\phi)$ with the property that ϕ is satisfiable if and only if $H(\phi)$ has a PMU cover of size r. Therefore, P3SAT reduces to PMUP. Since P3SAT is NP-complete [15], we conclude that PMUP is NP-complete even when restricted to planar bipartite graphs. □

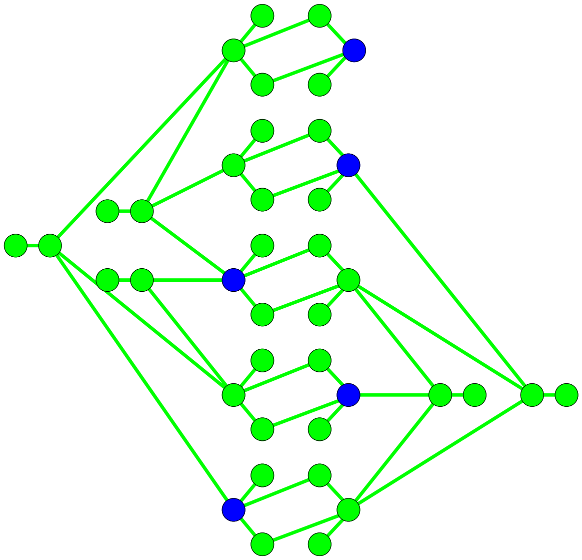


FIG. 13. *Minimum PMU cover (P3SAT with gadgets).*

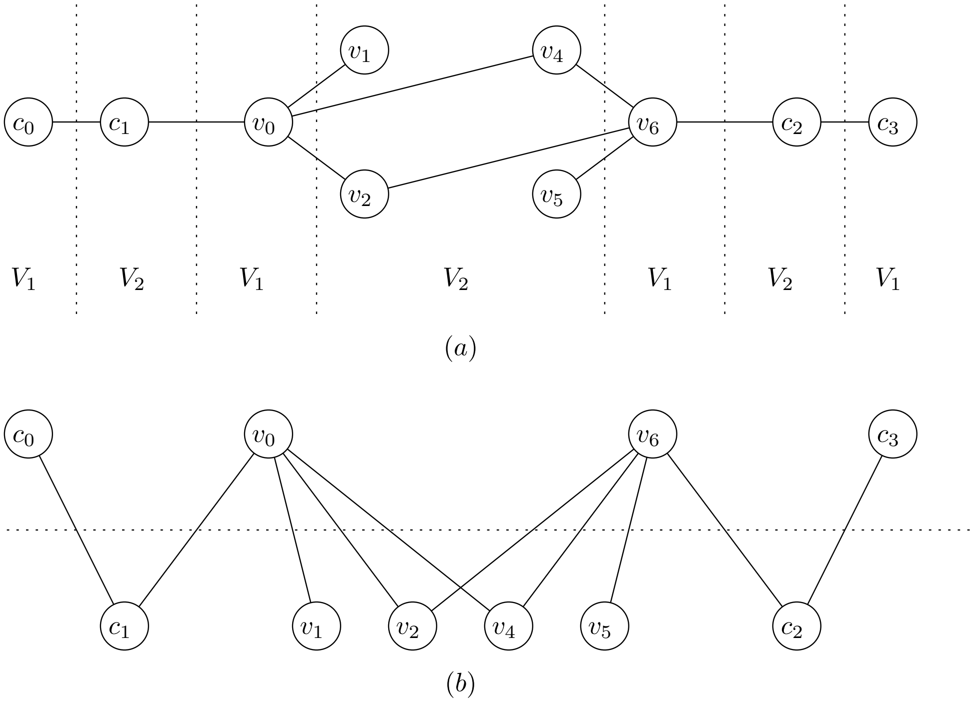


FIG. 14. *Partitions of nodes in $H(\phi)$ showing that $H(\phi)$ is bipartite.*

Acknowledgments. We wish to thank Monte Boisen and Donald Allison for helpful discussions. We also thank Lamine Mili and Thomas Baldwin for advice and contribution of test data. We extend our appreciation to the three referees who helped make this a better paper.

REFERENCES

[1] T. L. Baldwin, L. Mili, M. B. Boisen, Jr., and R. Adapa, *Power system observability with minimal phasor measurement placement*, IEEE Transactions on Power Systems, 8 (1993), pp. 707–715.

[2] D. J. Brueni, *Minimal PMU Placement for Graph Observability: A Decomposition Approach*, M.S. thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, 1993.

[3] R. O. Burnett, Jr., M. M. Butts, , T. W. Cease, V. Centeno, G. Michel, R. J. Murphy, and A. G. Phadke, *Synchronized phasor measurement of a power system event*, IEEE Transactions on Power Systems, 9 (1994), pp. 1643–1650.

[4] R. O. Burnett, Jr., M. M. Butts, and P. S. Sterlina, *Power system applications for phasor measurement units*, IEEE Computer Applications in Power, 7 (1994), pp. 8–13.

[5] C. S. Chen, C. W. Liu, and J. A. Jiang, *A new adaptive PMU based protection scheme for transposed/untransposed parallel transmission lines*, IEEE Transactions on Power Delivery, 17 (2002), pp. 395–404.

[6] S. M. El-Shal and J. S. Thorp, *Optimal placement of phasor measurement systems for enhancing on-line protection and control of large-scale power systems*, International J. Systems Sci., 21 (1990), pp. 1869–1880.

[7] M. R. Garey and D. S. Johnson, *Computers and Intractability*, W. H. Freeman and Company, San Francisco, 1979.

[8] I. O. Habiballah and M. R. Irving, *Observability analysis for state estimation using linear programming*, IEEE Proceedings—Generation, Transmission and Distribution, 148 (2001), pp. 142–145.

[9] T. W. Haynes, S. M. Hedetniemi, S. T. Hedetniemi, and M. A. Henning, *Domination in graphs applied to electric power networks*, SIAM J. Discrete Math., 15 (2002), pp. 519–529.

[10] L. S. Heath and A. L. Rosenberg, *Laying out graphs using queues*, SIAM J. Comput., 21 (1992), pp. 927–958.

[11] J. A. Jiang, Y. H. Lin, J. Z. Yang, T. M. Too, and C. W. Liu, *An adaptive PMU based fault detection/location technique for transmission lines—Part* II: *PMU implementation and performance evaluation*, IEEE Transactions on Power Delivery, 15 (2000), pp. 1136–1146.

[12] J. A. Jiang, J. Z. Yang, Y. H. Lin, C. W. Liu, and J. C. Ma, *An adaptive PMU based fault detection/location technique for transmission lines—Part I: PMU theory and algorithms*, IEEE Transactions on Power Delivery, 15 (2000), pp. 486–493.

[13] I. Kamwa and R. Grondin, *PMU configuration for system dynamic performance measurement in large multiarea power systems*, IEEE Transactions on Power Systems, 17 (2002), pp. 385–394.

[14] T. D. Liacco, *Real-time computer control of power systems*, Proceedings of the IEEE, 62 (1975), pp. 884–891.

[15] D. Lichtenstein, *Planar formulae and their uses*, SIAM J. Comput., 11 (1982), pp. 329–343.

[16] C. W. Liu, M. C. Su, S. S. Tsay, and J. Y. Wang, *Application of a novel fuzzy neural network to real-time transient stability swings prediction based on synchronized phasor measurements*, IEEE Transactions on Power Systems, 14 (1999), pp. 685–692.

[17] C.-W. Liu and J. S. Thorp, *Application of synchronized phasor measurements to real-time transient stability prediction*, IEE Proceedings—Generation, Transmission, and Distribution, 142 (1995), pp. 355–360.

[18] L. Mili, T. L. Baldwin, and R. Adapa, *Phasor measurement placement for voltage stability analysis of power systems*, in Proceedings of the 29th IEEE Conference on Decision and Control, Honolulu, HI, 1990, pp. 3033–3038.

[19] A. Monticelli, *Electric power system state estimation*, Proceedings of the IEEE, 88 (2000), pp. 262–282.

[20] E. W. Palmer and G. Ledwich, *Optimal placement of angle transducers in power systems*, IEEE Transactions on Power Systems, 11 (1996), pp. 788–793.

[21] A. G. Phadke, J. S. Thorp, and K. J. Karimi, *State estimation with phasor measurements*, IEEE Transactions on Power Systems, 1 (1986), pp. 233–241.

[22] A. Radovanovic, *Using the Internet in networking of synchronized phasor measurement units*, International Journal of Electrical Power and Energy Systems, 23 (2001), pp. 245–250.

[23] J. Sadeh, A. M. Ranjbar, N. Hadjsaid, and R. Feuillet, *Accurate fault location algorithm for power transmission lines*, European Transactions on Electrical Power, 10 (2000), pp. 313–318.

[24] R. E. Wilson and P. S. Sterlina, *GPS synchronized power system phase angle measurements*, International Journal of Satellite Communications, 12 (1994), pp. 499–505.

[25] C. S. Yu, C. W. Liu, S. L. Yu, and J. A. Jiang, *A new PMU-based fault location algorithm for series compensated lines*, IEEE Transactions on Power Delivery, 17 (2002), pp. 33–46.

SIAM J. Discrete Math.
Vol. 19, No. 3, pp. 762–797

ON THE EQUIVALENCE BETWEEN THE PRIMAL-DUAL SCHEMA AND THE LOCAL RATIO TECHNIQUE[*]

REUVEN BAR-YEHUDA[†] and DROR RAWITZ[‡]

Abstract. We discuss two approximation paradigms that were used to construct many approximation algorithms during the last two decades, the *primal-dual schema* and the *local ratio technique*. Recently, primal-dual algorithms were devised by first constructing a local ratio algorithm and then transforming it into a primal-dual algorithm. This was done in the case of the 2-approximation algorithms for the *feedback vertex set* problem and in the case of the first primal-dual algorithms for maximization problems. Subsequently, the nature of the connection between the two paradigms was posed as an open question by Williamson [*Math. Program.*, 91 (2002), pp. 447–478]. In this paper we answer this question by showing that the two paradigms are equivalent.

Key words. approximation algorithms, combinatorial optimization, covering problems, local ratio, primal-dual

AMS subject classifications. 68W25, 90C27

DOI. 10.1137/050625382

1. Introduction.

1.1. Primal-dual schema.

A key step in designing an approximation algorithm is establishing a good bound on the value of the optimum. This is where linear programming helps out. Many combinatorial optimization problems can be expressed as linear integer programs, and the value of an optimal solution to their *LP-relaxation* provides the desired bound. Clearly, the best we can hope for using this approach is to get an r-approximation algorithm, where r is the *integrality gap* of the program. One way to obtain approximate solutions is to solve the LP-relaxation and then to *round* the solution while ensuring that the cost does not change by much. Another way to go about it is to use the *dual* of the LP-relaxation in the design of approximation algorithms and their analyses. A *primal-dual* r-approximation algorithm constructs a feasible integral primal solution and a feasible dual solution such that the value of the primal solution is no more than r times (or, in the maximization case, at least $1/r$ times) the value of the dual solution. This work focuses on classical primal-dual approximation algorithms, specifically those that fall within the so-called *primal-dual schema*.

The primal-dual schema can be seen as a modified version of the *primal-dual method for solving linear programs*. The primal-dual method was originally proposed by Dantzig, Ford, and Fulkerson [19]. Over the years, it became an important tool for solving combinatorial optimization problems that can be formulated as linear programs. While the *complementary slackness conditions* are imposed in the primal-dual

[*] Received by the editors February 26, 2005; accepted for publication (in revised form) March 28, 2005; published electronically December 7, 2005. An extended abstract containing some of these results was presented at the 4th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'01) [10].

http://www.siam.org/journals/sidma/19-3/62538.html

[†] Department of Computer Science, Technion, Haifa 32000, Israel (reuven@cs.technion.ac.il). The research of this author was supported by the N. Haar and R. Zinn Research Fund.

[‡] School of Electrical Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel (rawitz@eng.tau.ac.il). This research was done while the author was a Ph.D. student at the Computer Science Department of the Technion.

method, we enforce the primal conditions and relax the dual conditions when working with the primal-dual schema. A primal-dual approximation algorithm typically constructs an approximate primal solution and a feasible dual solution simultaneously. The approximation ratio is derived from comparing the values of both solutions. The first approximation algorithm to use the primal-dual schema is Bar-Yehuda and Even's approximation algorithm for the *weighted set cover* problem [6], and since then many approximations algorithms for NP-hard optimization problems were constructed using this approach, among which are algorithms for *network design* problems (see, e.g., [37, 1, 26]). In fact, this line of research has introduced the idea of looking at *minimal* solutions (with respect to set inclusion) to the primal-dual schema.

Several primal-dual approximation frameworks were proposed in the last decade. Goemans and Williamson [26] presented a generic algorithm for a wide family of *network design* problems. They based a subsequent survey of the primal-dual schema [27] on this algorithm. Another, more recent, survey by Williamson [39] describes the primal-dual schema and several extensions of the primal-dual approach. In [27] the authors show that the primal-dual schema can be used to explain many classical (exact and approximation) algorithms for special cases of the *hitting set* problem, such as *shortest path*, *minimum spanning tree*, and *vertex cover*. Following [26], Bertsimas and Teo [14] proposed a primal-dual framework to design and analyze approximation algorithms for integer programming problems of the covering type. As in [26, 27] this framework enforces the primal complementary slackness conditions while relaxing the dual conditions. However, in contrast to previous studies, Bertsimas and Teo [14] express each advancement step as the construction of a single valid inequality and an increase of the corresponding dual variable (as opposed to an increase of several dual variables). The approximation ratio of the resulting algorithm depends upon the quality, or *strength* in terms of [14], of the inequalities that are used.

1.2. Local ratio technique. The local ratio technique uses weight subtractions. An advancement step of a local ratio algorithm typically consists of the construction of a new *weight function*, which is then subtracted from the current objective function. Each subtraction changes the optimum, but incurs a cost. The ratio between this cost and the change in the optimum is called the *effectiveness* of the weight function. The approximation ratio of a local ratio algorithm depends on the effectiveness of the weight functions it constructs.

The local ratio approach was developed by Bar-Yehuda and Even [7] in order to approximate the *set cover* and *vertex cover* problems. In that paper the authors presented a local ratio analysis to their primal-dual approximation algorithm for *set cover* [6] and a $(2 - \frac{\log \log n}{2 \log n})$-approximation algorithm for vertex cover. About ten years later Bafna, Berman, and Fujito [2] extended the *local ratio lemma* from [7] in order to construct a 2-approximation algorithm for the *feedback vertex set* problem. This algorithm was the first local ratio algorithm that used the notion of *minimal* solutions. We note that this work and the 2-approximation from [13] were essential in the design of primal-dual approximation algorithms for *feedback vertex set* [17]. Following Bafna, Berman, and Fujito [2], Fujito [23] presented a generic local ratio algorithm for node deletion problems with nontrivial and hereditary graph properties.[1] Later, Bar-Yehuda [4] presented a unified local ratio approach for developing and analyzing approximation algorithms for covering problems. This framework, which extends

[1] A graph property π is *nontrivial* if it is true for infinitely many graphs and false for infinitely many graphs; it is *hereditary* if every subgraph of a graph satisfying π also satisfies π.

the one in [23], can be used to explain most known optimization and approximation algorithms for covering problems. Bar-Noy et al. [3] use the local ratio technique to develop a framework for resource allocation and scheduling problems. This study was the first to present a local ratio (or primal-dual) approximation algorithm for a natural maximization problem. A primal-dual interpretation was presented in [3] as well. Recently, Bar-Yehuda and Rawitz [11] presented local ratio interpretations of known algorithms for *minimum s-t cut* and the *assignment* problem. These algorithms are the first applications of local ratio to use negative weights. The corresponding primal-dual analyses are based on new integer programming formulations of these fundamental problems. A detailed survey on the local ratio technique that includes recent developments (such as fractional local ratio [8]) is given in [5].

1.3. Our results. We present two generic approximation algorithms for covering problems. The first is a recursive version of the primal-dual algorithm from [14], and the second is a variant of the local ratio algorithm from [4]. After presenting both frameworks we discuss the connection between them. We show that a *strong* valid inequality (in terms of [14]) and an *effective* weight function (in terms of [4]) are equivalent notions. Consequently, we prove that both frameworks for covering are one and the same. We demonstrate the combined approach on a variety of covering problems, such as *network design* problems and the *feedback vertex set* problem. We also present a linear time approximation algorithm for the *generalized hitting set* problem (which can be viewed as a prize-collecting version of hitting set). This algorithm extends the approximation algorithm for hitting set from [6] and achieves a ratio of 2 in the special case of *generalized vertex cover.* Its time complexity is significantly better than Hochbaum's [31] $\mathrm{O}(nm \log \frac{n^2}{m})$ 2-approximation algorithm for this special case.

Next, we extend both our frameworks to include algorithms for minimization problems that are not covered by the generic algorithms from [14] and [4]. We show that the equivalence between the paradigms continues to hold. We demonstrate the use of the extended frameworks on several algorithms: a 2.5-approximation algorithm for *feedback vertex set in tournaments* [16]; a 2-approximation algorithm for a noncovering problem called *minimum* 2*-satisfiability* [29, 9]; and a 3-approximation algorithm for a *bandwidth trading* problem [15]. We show that the equivalence continues to hold in the maximization case. We do that by developing two equivalent frameworks for maximization problems, one in each approach. Algorithms for *interval scheduling* [3] and *longest path in a* directed acyclic graph (DAG) are used to demonstrate our maximization frameworks.

It is important to note that the equivalence between the paradigms is constructive. That is, a primal-dual algorithm that follows our framework can be easily transformed into a local ratio algorithm, and vice versa. A corollary to this equivalence is that the *integrality gap* of a certain integer program serves as a lower bound to the approximation ratio of a local ratio algorithm. We also note that the nature of the connection between the two paradigms was mentioned as an open question by Williamson [39].

We believe that this study contributes to the understanding of both approaches and, especially, that it may help in the design of approximation algorithms for noncovering problems and nonstandard algorithms for covering problems. For example, we show that the primal-dual schema can be applied as a clean-up phase whose output is an instance of a certain type that we know how to solve by other means. This approach is quite natural in the local ratio setting and has been used in the $(2 - \frac{\log \log n}{2 \log n})$-approximation algorithm for vertex cover [7] and the 2.5-approximation algorithm for feedback vertex set in tournaments [16].

1.4. Related work. Jain and Vazirani [34] presented a 3-approximation algorithm for the *metric uncapacitated facility location* (MUFL) problem that deviates from the standard primal-dual paradigm. Their algorithm does not employ the usual mechanism of relaxing the dual complementary slackness conditions, but rather it relaxes the *primal* conditions. Jain et al. [33] developed *dual fitting* algorithms for MUFL. A dual fitting algorithm produces a feasible primal solution and an *infeasible* dual solution such that (1) the cost of the dual solution dominates the cost of the primal solution, and (2) dividing the dual solution by an appropriately chosen r results in a feasible dual solution. These two properties imply that the primal solution is r-approximate. This contrasts with the standard primal-dual approach, in which a feasible dual solution is found and used to direct the construction of a primal solution. Freund and Rawitz [22] presented two combinatorial approximation frameworks that are not based on LP-duality. Instead, they are based on weight manipulation in the spirit of the local ratio technique. They showed that the first framework is equivalent to *dual fitting* and that the second framework is equivalent to a linear programming–based method which they defined and called *primal fitting*. The second framework can be used to analyze the algorithm of Jain and Vazirani [34].

1.5. Overview. The remainder of the paper is organized as follows. In section 2 we define the family of problems which we consider in this paper and state some basic facts regarding primal-dual and local ratio. In section 3 we demonstrate the two approaches on the *Steiner tree* problem. The objective of this example is to identify the differences and similarities between the paradigms. Section 4 discusses *covering* problems. We present a generic primal-dual algorithm and a generic local ratio algorithm, both for covering problems, and we show that they are equivalent. We also show how the two generic algorithms can be applied to several covering problems. More general minimization frameworks are given in section 5, and our maximization frameworks are given in section 6.

2. Preliminaries.

We consider the following optimization problem: given a nonnegative *weight* vector $w \in \mathbb{R}^n_+$, find a solution $x \in \mathbb{N}^n$ that minimizes (or maximizes) the inner product $w \cdot x$ subject to some set $\mathcal{F}$ of feasibility constraints on x. This formulation contains, among others, all linear and integer programming problems. Usually, we require $x \in \{0,1\}^n$, and in this case we abuse notation by treating a vector $x \in \{0,1\}^n$ as the set of its 1 entries, i.e., as $\{j : x_j = 1\}$. The correct interpretation should be clear from the context.

We define the following for a minimization (maximization) problem $(\mathcal{F}, w)$. A vector x is called a *feasible solution* if x satisfies the constraints in $\mathcal{F}$. A feasible solution x^* is *optimal* if every feasible solution x satisfies $w \cdot x^* \leq w \cdot x$ $(w \cdot x^* \geq w \cdot x)$. We denote by Opt the value of an optimal solution, i.e., the optimum value. A feasible solution x is called an *r-approximation* or *r-approximate* if $w \cdot x \leq r \cdot w \cdot x^*$ $(w \cdot x \geq \frac{1}{r} \cdot w \cdot x^*)$, where x^* is an optimal solution. An algorithm is called an *r-approximation algorithm* if it returns r-approximate solutions. Namely, an r-approximation algorithm returns a feasible solution whose weight is no more than r (at least $1/r$) times the optimum weight.

2.1. Primal-dual. This section is written in terms of minimization problems. Similar arguments can be given in the maximization case. Also, in what follows we assume basic knowledge of linear programming. (See, e.g., [36, 35] for more details about linear programming.)

Consider the linear program

$$\begin{array}{lll} \min & \displaystyle\sum_{j=1}^{n} w_j x_j & \\ \text{s.t.} & \displaystyle\sum_{j=1}^{n} a_{ij} x_j \geq b_i & \forall i \in \{1, \ldots, m\}, \\ & x_j \geq 0 & \forall j \in \{1, \ldots, n\} \end{array}$$

and its dual

$$\begin{array}{lll} \max & \displaystyle\sum_{i=1}^{n} b_i y_i & \\ \text{s.t.} & \displaystyle\sum_{i=1}^{n} a_{ij} y_i \leq w_j & \forall j \in \{1, \ldots, n\}, \\ & y_i \geq 0 & \forall i \in \{1, \ldots, m\}. \end{array}$$

A primal-dual r-approximation algorithm for a minimization problem produces an integral primal solution x and a dual solution y such that the weight of the primal solution is no more than r times the value of dual solution. Namely, it produces an integral solution x and a solution y such that

$$wx \leq r \cdot by. \tag{2.1}$$

The weak duality theorem implies that x is r-approximate.

One way to design an algorithm that finds a pair of primal and dual solutions that satisfies (2.1) is to restrict our attention to a specific kind of pairs of primal and dual solutions. Consider a primal solution x and a dual solution y. The duality theorem provides us with a way to characterize a pair of optimal solutions. Specifically, x and y are optimal if and only if the following conditions, called the *complementary slackness conditions*, are satisfied:

$$\begin{array}{lll} \text{Primal conditions:} & \forall j,\ x_j > 0 & \Rightarrow \displaystyle\sum_{i=1}^{m} a_{ij} y_i = w_j. \\ \text{Dual conditions:} & \forall i,\ y_i > 0 & \Rightarrow \displaystyle\sum_{j=1}^{n} a_{ij} x_j = b_i. \end{array}$$

However, we are interested in approximate solutions, and thus it seems natural to relax the complementary slackness conditions. Consider an integral primal solution x and a dual solution y that satisfy the following conditions, called the *relaxed complementary slackness conditions* [38]:

$$\begin{array}{lll} \text{Relaxed primal conditions:} & \forall j,\ x_j > 0 & \Rightarrow w_j / r_1 \leq \displaystyle\sum_{i=1}^{m} a_{ij} y_i \leq w_j. \\ \text{Relaxed dual conditions:} & \forall i,\ y_i > 0 & \Rightarrow b_i \leq \displaystyle\sum_{j=1}^{n} a_{ij} x_j \leq r_2 \cdot b_i. \end{array}$$

Then

$$\sum_{j=1}^{n} w_j x_j \leq \sum_{j=1}^{n} r_1 \cdot \left(\sum_{i=1}^{m} a_{ij} y_i \right) x_j = r_1 \cdot \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j \right) y_i \leq r_1 \cdot r_2 \cdot \sum_{i=1}^{m} b_i y_i,$$

which means that x is $r_1 \cdot r_2$-approximate.

In this study we consider algorithms in which $r_1 = 1$, that is, algorithms that relax only the dual complementary slackness conditions. (Algorithms that relax the primal conditions are studied in [22].) Typically, such an algorithm constructs an integral primal solution x and a feasible dual solution y simultaneously. It starts with an infeasible primal solution and a feasible dual solution (usually, $x = 0$ and $y = 0$). It iteratively raises the dual solution and improves the feasibility of the primal solution. In each iteration the dual solution is increased while ensuring that the relaxed dual conditions are satisfied. Also, a primal variable can be increased only if its corresponding primal condition is obeyed.

2.2. Local ratio. Say we want to construct an r-approximation algorithm for a minimization problem. A key step in the design of such an algorithm is to establish a good lower bound b on the weight of the optimal solution. This bound can later be used in the analysis to prove that the solution found by the algorithm is r-approximate by showing that its weight is no more than $r \cdot b$. The local ratio technique uses a "local" variation of this idea. In essence, the idea is to break down the weight w of the solution found by the algorithm into a sum of "partial weights" $w = w_1 + w_2 + \cdots + w_k$, and similarly break down the lower bound b into $b = b_1 + b_2 + \cdots + b_k$, and to show that $w_i \leq r \cdot b_i$ for all i. The breakdown of w and b is determined by the manner in which the solution is constructed by the algorithm. In fact, the algorithm constructs the solution in such a way as to ensure that such a breakdown exists. Put differently, at the ith step, the algorithm "pays" $r \cdot b_i$ and manipulates the problem instance so that the optimum drops by at least b_i.

The local ratio technique is based on the following theorem. (The proof is given for completeness.)

Theorem 2.1 (local ratio theorem [3]). *Let $(\mathcal{F}, w)$ be a minimization (maximization) problem, and let w, w_1, and w_2 be weight functions such that $w = w_1 + w_2$. Then if x is r-approximate with respect to $(\mathcal{F}, w_1)$ and with respect to $(\mathcal{F}, w_2)$, then x is r-approximate with respect to $(\mathcal{F}, w)$.*

Proof. Let x^*, x_1^*, x_2^* be optimal solutions with respect to $(\mathcal{F}, w), (\mathcal{F}, w_1)$, and $(\mathcal{F}, w_2)$, respectively. Then in the minimization case we have

$$wx = w_1x + w_2x \leq r \cdot w_1x_1^* + r \cdot w_2x_2^* \leq r \cdot (w_1x^* + w_2x^*) = r \cdot wx^* \ .$$

For the maximization case simply replace $\leq$ by $\geq$ and r by $\frac{1}{r}$. □

Note that $\mathcal{F}$ can include arbitrary feasibility constraints, and not just linear, or linear integer, constraints. Nevertheless, all successful applications of the local ratio technique to date involve problems in which the constraints are linear.

Usually, the local ratio theorem is used in the following manner. Given a problem instance with a weight function w, we find a nonnegative weight function $\delta \leq w$ such that every *minimal* solution (with respect to set inclusion) is r-approximate with respect to δ. Then we recursively find a minimal solution that is r-approximate with respect to $w - \delta$. By the local ratio theorem this solution is r-approximate with respect to the original weights w. The recursion terminates when a minimal r-approximate solution can be found directly, which usually occurs when the problem instance is an empty instance, or when the weights have evolved to the point that the set of all zero-weight elements constitutes a feasible (and hence optimal) solution. Note that the scheme just described is tail recursive and can thus be implemented iteratively rather than recursively.

3. An introductory example: The Steiner tree problem. In this section we compare two approximation algorithms for the *Steiner tree* problem, one based on the primal-dual schema and the other on the local ratio technique. The algorithms are not new, but they demonstrate how one usually uses both paradigms and thus help us to identify differences and similarities between the two approaches. Also, this example will be useful in the next section. We start with the definition of the problem.

Given a graph $G = (V, E)$ and a nonempty set of *terminals* $T \subseteq V$, a *Steiner tree* is a subtree of G that connects all the vertices in T. Given a nonnegative weight function w on the edges, the Steiner tree problem is to find a minimum weight Steiner tree, where the weight of a tree is the total weight of its edges. (We consider trees to be sets of edges.)

We are interested in Steiner trees that are *minimal* with respect to set inclusion. Namely, a Steiner tree F is minimal if $F \setminus \{e\}$ is not a Steiner tree for every edge $e \in F$. Observe that a Steiner tree is minimal if and only if every leaf in the tree is a terminal. For an edge $e \in E$ we denote the number of terminals incident to e, or the *terminal degree* of e, by $\tau(e)$, i.e., $\tau(e) = |e \cap T|$.

LEMMA 3.1. *Let F be a minimal Steiner tree. Then* $|T| \leq \sum_{e\in F} \tau(e) \leq 2\,|T| - 2$.

Proof. The first inequality follows from the fact that every terminal must be incident to some edge in F. The second inequality can be proven as follows. We pick an arbitrary terminal r to be the root of the Steiner tree. Next, we place a total of $2\,|T| - 2$ coins on the terminals—two coins on each terminal in $T \setminus \{r\}$—and show that we can reassign the coins such that there are at least $\tau(e)$ coins on each edge $e \in F$. Consider a terminal $t \in T \setminus \{r\}$, and let u be the parent of t. Let s be the terminal which is closest to u on the path from u to r, and let v be s's child on that path. t places one coin on the edge (t, u) and another coin on the edge (v, s). (If $u = s$ and $v = t$, then two coins are placed on (t, u).) It is not hard to verify that, because the leaves of F are terminals, at least $\tau(e)$ coins are placed on every edge $e \in F$. ☐

A slightly different proof of a more general claim is given in [27].

3.1. Primal-dual. A typical first step in the design of a primal-dual approximation algorithm is to find a suitable formulation of the problem at hand as a linear integer program. Indeed, we start with such a formulation of the Steiner tree problem. We say that a subset $S \subseteq V$ *splits* T if $\emptyset \subsetneq S \cap T \subsetneq T$. Let $\textsc{Split}(T)$ be the set of all subsets of V that split T, i.e., $\textsc{Split}(T) = \{S : \emptyset \subsetneq S \cap T \subsetneq T\}$. The Steiner tree problem can be formulated by the following linear integer program:

$$\text{(ST)} \qquad \begin{array}{lll} \min & \sum_{e\in E} w(e)x_e & \\ \text{s.t.} & \sum_{e\in(S,\bar{S})} x_e \geq 1 & \forall S \in \textsc{Split}(T), \\ & x_e \in \{0,1\} & \forall e \in E, \end{array}$$

where $(S, \bar{S})$ denotes the set of edges having exactly one endpoint in S. We get an LP-relaxation by replacing the last set of constraints by $x_e \geq 0$ for all $e \in E$. The corresponding dual program is

$$\begin{array}{lll} \max & \sum_{S\in\textsc{Split}(T)} y_S & \\ \text{s.t.} & \sum_{S:e\in(S,\bar{S})} y_S \leq w(e) & \forall e \in E, \\ & y_S \geq 0 & \forall S \in \textsc{Split}(T). \end{array}$$

```
Algorithm PD-ST$(G, w)$.
1.  $F \leftarrow \emptyset$
2.  $y \leftarrow 0$
3.  $\mathcal{C}_0 \leftarrow \{\{v\} : v \in V\}$
4.  $\ell \leftarrow 0$
5.  While $\exists C \in \mathcal{C}_\ell$ such that $C$ splits $T$
6.      $\ell \leftarrow \ell + 1$
7.      Increase $y_C$ uniformly for every $C \in \mathcal{C}$ that splits $T$
        until some dual constraint becomes tight
8.      Let $e_\ell = (u, v)$, such that $u \in C_i$ and $v \in C_j$,
        be an edge that corresponds to a tight dual constraint
9.      $F \leftarrow F \cup \{e_\ell\}$
10.     $\mathcal{C}_\ell \leftarrow \mathcal{C}_{\ell-1} \cup \{C_i \cup C_j\} \setminus \{C_i, C_j\}$
11. For $j \leftarrow \ell$ down-to 1
12.     If $F \setminus \{e_j\}$ is feasible then $F \leftarrow F \setminus \{e_j\}$
13. Output $F$
```

FIG. 3.1.

Algorithm **PD-ST** is a primal-dual approximation algorithm for the Steiner tree problem (see Figure 3.1). It is a specific implementation of the generic algorithm from [26]. The algorithm starts with $|V|$ components—each containing a single vertex. The components are induced by the solution F. In the ℓth iteration it raises the dual variables that correspond to components that split T until some dual constraint becomes tight. Then an edge that corresponds to some tight dual constraint is added to F, and the components are updated accordingly. This process terminates when all terminals are in the same component. Then F is turned into a minimal Steiner tree using reverse deletion.

First, we show that Algorithm **PD-ST** produces feasible solutions. Consider a solution F returned by the algorithm. Observe that all the terminals are in the same component; otherwise the algorithm would not have terminated. Also, due to lines 11–12 F is a minimal Steiner tree.

We need only prove that Algorithm **PD-ST** produces 2-approximate solutions. Let y be the dual solution corresponding to a solution F that was output by the algorithm. By the weak duality theorem $\sum_{S \in \text{SPLIT}(T)} y_S \leq \text{OPT}$. Thus, in order to show that F is 2-approximate, it is enough to prove that $\sum_{e \in F} w(e) \leq 2 \cdot \sum_{S \in \text{SPLIT}(T)} y_S$.

In the ℓth iteration the algorithm raises y_C for every component C that splits T, and therefore

$$\sum_{S \in \text{SPLIT}(T)} y_S = \sum_{\ell=1}^{t} \epsilon_\ell \left|\mathcal{C}'_\ell\right| ,$$

where ϵ_ℓ is the dual increase at the ℓth iteration, and $\mathcal{C}'_\ell \subseteq \mathcal{C}_\ell$ is the set of components that split T (*active* components in the terminology of [26]). On the other hand, only edges that correspond to tight dual constraints are taken into the solution F, and hence

$$\sum_{e \in F} w(e) = \sum_{e \in F} \sum_{S : e \in (S, \bar{S})} y_S = \sum_{e \in F} \sum_{S : e \in (S, \bar{S})} \sum_{\ell : S \in \mathcal{C}'_\ell} \epsilon_\ell = \sum_{\ell=1}^{t} \epsilon_\ell \sum_{C \in \mathcal{C}'_\ell} \left|(C, \bar{C}) \cap F\right| .$$

Thus, it is enough to prove that for every $\ell \in \{1,\dots,t\}$,

$$\sum_{C\in\mathcal{C}'_\ell} \left|(C,\bar{C})\cap F\right| \leq 2\cdot|\mathcal{C}'_\ell| \ .$$

Observe that for a component $C\in\mathcal{C}'_\ell$, $\left|(C,\bar{C})\cap F\right|$ is the number of edges in F with one endpoint in C. If we could prove that $\left|(C,\bar{C})\cap F\right| \leq 2$ for every $C\in\mathcal{C}'_\ell$, then we are done. However, this is not necessarily true. Instead, we prove an "amortized" version of this claim. That is, we prove that the average number of edges in F with one endpoint in a component $C\in\mathcal{C}'_\ell$ is no more that two. We remark that by doing that we actually prove that the relaxed dual complementary slackness conditions are satisfied (as shown in the next section).

Consider the ℓth iteration, and define a multigraph (a graph that may contain multiple edges between pairs of vertices) $G^\ell = (V^\ell, E^\ell)$ as follows. Each vertex in V^ℓ corresponds to a component $C\in\mathcal{C}_\ell$. We refer to a vertex u as a terminal in G^ℓ if the corresponding component in G contains at least one terminal (i.e., if the corresponding component is in $\mathcal{C}'_\ell$). We denote the set of terminals in G^ℓ by T^ℓ. Let u and v be vertices in G^ℓ and let C_u and C_v be the corresponding components. E^ℓ contains a copy of the edge (u,v) for every edge $(x,y)\in E$ such that $x\in C_u, y\in C_v$, and the weight of this copy is $w(x,y)$. Consider the set of edges F^ℓ that is induced by F in G^ℓ. Clearly,

$$\sum_{C\in\mathcal{C}'_\ell} \left|(C,\bar{C})\cap F\right| = \sum_{v\in T^\ell} \left|E^\ell(v)\cap F^\ell\right| = \sum_{e\in F^\ell} \tau_{G^\ell}(e),$$

where $E^\ell(v)$ is the set of edges incident on v (in G^ℓ). We claim that F^ℓ is a minimal Steiner tree in G^ℓ. To see this observe that in the ℓth iteration the terminals in each component C are connected in G (by edges within each component). Moreover, due to the reverse deletion phase (lines 11–12) the edges in F^ℓ form a minimal Steiner tree in G^ℓ. Thus, by Lemma 3.1, we know that

$$\sum_{e\in F^\ell} \tau_{G^\ell}(e) \leq 2\cdot|T^\ell| - 2 = 2\cdot|C'_\ell| - 2$$

and we are done.

3.2. Local ratio. The following local ratio approximation algorithm (see Figure 3.2) appeared in [4] (though in less detail). In the course of its execution, the algorithm modifies the graph by performing *edge contractions*. Contracting an edge (u,v) consists of "fusing" its two endpoints u and v into a single (new) vertex z. The edge connecting u and v is deleted and every other edge incident on u or v becomes incident on z instead. In addition, if either u or v are terminals, then z is a terminal too.

Note the slight abuse of notation in line 7. The weight function in the recursive call is not $w-\delta$ itself, but rather the restriction on G'. We will continue to silently abuse notation in this manner.

We prove by induction on the number of terminals that Algorithm **LR-ST** returns a minimal Steiner tree. At the recursion basis the solution returned is the empty set, which is both feasible and minimal. For the inductive step, by the inductive hypothesis, F' is a minimal Steiner tree with respect to G' and T'. Since we add e to F only if we have to, F is a minimal Steiner tree with respect to G and T.

```
Algorithm LR-ST$(G, T, w)$.
    If $G$ contains only one terminal then return $\emptyset$
    Else:
        Let $\epsilon = \min_e \{w(e)/\tau(e)\}$
        Define the weight function $\delta(e) = \epsilon \cdot \tau(e)$
        Let $e$ be an edge such that $w(e) = \delta(e)$
        Let $(G', T')$ be the instance obtained by contracting $e$
        $F' \leftarrow$ LR-ST$(G', T', w - \delta)$
        If $F'$ is not feasible then return $F = F' \cup \{e\}$
        Else, return $F = F'$
```

Fig. 3.2.

It remains to prove that Algorithm **LR-ST** produces 2-approximate solutions. The proof is also by induction on the number of terminals. In the base case the solution returned is the empty set, which is optimal. For the inductive step, by the inductive hypothesis, F' is 2-approximate with respect to G', T', and $w - \delta$. Since $(w - \delta)(e) = 0$, the weight of F with respect to $w - \delta$ equals that of F', and the optimum value for G, T with respect to $w - \delta$ cannot be smaller than the optimum value for G', T', because if F^* is an optimal solution for G, T, then $F^* \setminus \{e\}$ is a feasible solution of the same weight for G', T'. Thus, F is 2-approximate with respect to G, T, and $w - \delta$. By Lemma 3.1, any minimal Steiner tree in G is 2-approximate with respect to δ. Thus, by the local ratio theorem, F is 2-approximate with respect to G, T, and w as well.

3.3. Primal-dual vs. local ratio. Algorithms **PD-ST** and **LR-ST** represent many algorithms in the literature in the sense that each of them can be viewed as a standard use of the corresponding paradigm. Algorithm **PD-ST** relies heavily on LP-duality. It is based on a predetermined linear program and its dual program, and its analysis is based on the comparison between the values of an integral primal solution and a dual solution. Algorithm **PD-ST** is iterative, and in each iteration the dual solution is changed. In a sense, the dual solution can be viewed as the bookkeeper of the algorithm. On the other hand, Algorithm **LR-ST** does not use linear programming. Instead, it relies upon weight decompositions and a local ratio theorem. As in this case, local ratio algorithms are typically recursive, and in each recursive call the weights are decomposed and the instance is modified. The decomposition is determined by a weight function defined in the current recursive call. Thus, at least at first glance, the two algorithms and their analyses seem very different.

Having said all that, we turn to the similarities between the algorithms. Both algorithms use the same combinatorial property (Lemma 3.1) to achieve an approximate solution. The performance ratio of both algorithms was proven locally. That is, it was shown, using the above-mentioned property, that in each iteration/decomposition a certain ratio holds. Also, both solutions use a reverse deletion phase. In the next section we show that this is no coincidence. The equivalence between the paradigms is based on the fact that "good" valid inequalities are equivalent to "good" weight functions. We shall also see that the changes in the dual during a primal-dual algorithm are strongly connected to the values of ϵ that are chosen in the recursive calls of a local ratio algorithm.

4. Covering problems. Perhaps the most famous covering problem is the *set cover* problem. In this problem we are given a collection of sets $\mathcal{C} = \{S_1, \ldots, S_m\}$ and a weight function w on the sets. The objective is to find a minimum-weight collection of sets that *covers* all elements. In other words, a collection $\mathcal{C}' \subseteq \mathcal{C}$ is a *set cover* if each element in $\bigcup_{i=1}^{m} S_i$ is contained in some set from $\mathcal{C}'$, and we aim to find a set cover of minimum weight. Consider a set cover $\mathcal{C}'$. Clearly, if we add sets from $\mathcal{C} \setminus \mathcal{C}'$ to $\mathcal{C}'$, the resulting collection is also a set cover. This property is shared by all *covering problems.* A minimization problem $(\mathcal{F}, w)$ is called a covering problem if (1) $x \in \{0,1\}^n$, and (2) any extension of a feasible solution to any possible instance is always feasible. In this case, we call the set of constraints $\mathcal{F}$ *monotone.* Note that a monotone set of linear constraints typically contains inequalities with nonnegative coefficients.

The family of covering problems contains a broad range of optimization problems. Many of them, such as *vertex cover*, *feedback vertex set*, and *Steiner tree*, were studied extensively. In fact, both the primal-dual schema and the local ratio technique were developed for the purpose of finding good approximate solutions for the *set cover* problem and its special case, the *vertex cover* problem.

Primal-dual approximation algorithms for covering problems traditionally reduce the size of the instance at hand in each iteration by adding an element $j \in \{1, \ldots, n\}$ whose corresponding dual constraint is tight to the primal solution (see, e.g., [27, 14]). Local ratio algorithms for covering problems implicitly add all zero-weight elements to the solution and, therefore, reduce the size of the instance in each step as well (see, e.g., [4]). In order to implement this we alter the problem definition by adding a set (or vector), denoted by z, which includes elements that are considered (at least, temporarily) to be taken into the solution. This makes it easier to present primal-dual algorithms recursively and to present local ratio algorithms in which the addition of zero-weight elements to the partial solution is explicit.

More formally, given a monotone set of constraints $\mathcal{F}$, a weight function w, and a vector $z \in \{0,1\}^n$, we are interested in the following problem. Find a vector $x \in \{0,1\}^n$ such that (1) $z \cap x = \emptyset$, (2) $x \cup z$ satisfies $\mathcal{F}$, and (3) x minimizes the inner product $w \cdot x$. (When $z = \emptyset$ we get the original problem $(\mathcal{F}, w)$.) z can be viewed as an additional monotone constraint, and therefore this problem is a covering problem. The definitions of a feasible solution, an optimal solution, and an r-approximate solution can be understood in a straightforward manner. We denote the set of feasible solutions with respect to $\mathcal{F}$ and z by $\textsc{Sol}(\mathcal{F}, z)$. Also, a feasible solution x is called *minimal* (with respect to set inclusion) if for all $j \in x$ the vector $z \cup x \setminus \{j\}$ is not feasible.

We remark that the use of this terminology is very useful in the context of this paper, i.e., for presenting generic algorithms, and for showing the equivalence between the two paradigms. However, it may be inept at constructing an approximation algorithm for a specific problem.

4.1. A primal-dual framework for covering problems. In this section we present a recursive primal-dual framework for approximating covering problems that is based on the one by Bertsimas and Teo [14]. However, before doing so we show that the framework from [14] extends the generic algorithm of Goemans and Williamson [27]. The proof of this claim is based on the observation that every advancement step of an approximation algorithm that uses the primal-dual schema can be represented by a change in a single dual variable. Note that this was not shown explicitly in [14] and was also mentioned by Williamson [39]. The reason we show this explicitly is twofold. First, we would like to draw attention to the fact that most primal-dual algorithms

Algorithm GW.

```
$y \leftarrow 0$
$x \leftarrow \emptyset$
$j \leftarrow 0$
While $x$ is not feasible
    $j \leftarrow j + 1$
    $\mathcal{V}_j \leftarrow$ VIOLATION$(x)$
    Increase $y_k$ uniformly for all $T_k \in \mathcal{V}_j$
            until $\exists e_j \notin x : \sum_{i:e_\ell \in T_i} y_i = w_{e_j}$
    $x \leftarrow x \cup \{e_j\}$
$\ell \leftarrow j$
For $j \leftarrow \ell$ down-to 1
    If $x \setminus \{e_j\}$ is feasible then $x \leftarrow x \setminus \{e_j\}$
Output $x$
```

FIG. 4.1.

in the literature do not follow the framework from [14], and therefore their analyses are unnecessarily complicated and do not offer much insight into the design process of the algorithm. (This is in contrast to local ratio analyses.) Second, we want to make the role of the complementary slackness conditions in primal-dual analyses more apparent.

We start by presenting the algorithm of Goemans and Williamson [27, p. 158]; see Figure 4.1. The algorithm and its analysis are included for completeness. Goemans and Williamson base their generic algorithm on the *hitting set* problem. In this problem we are given a collection of subsets $T_1, \ldots, T_q$ of a ground set E and a weight function $w : E \to \mathbb{R}_+$. Our goal is to find a minimum weight subset $x \subseteq E$ such that $x \cap T_i \neq \emptyset$ for every $i \in \{1, \ldots, q\}$. In turns out that many known problems (shortest path, vertex cover, etc.) are special cases of the hitting set problem. The hitting set problem can be formulated as follows:

$$\begin{array}{lll} \min & \displaystyle\sum_{e \in E} w_e x_e & \\ \text{s.t.} & \displaystyle\sum_{e \in T_i} x_e \geq 1 & \forall i \in \{1, \ldots, q\}, \\ & x_e \in \{0, 1\} & \forall e \in E, \end{array}$$

where $x_e = 1$ if and only if $e \in x$. The LP-relaxation and the corresponding dual program are

$$\begin{array}{lll} \min & \displaystyle\sum_{e \in E} w_e x_e & \\ \text{s.t.} & \displaystyle\sum_{e \in T_i} x_e \geq 1 & \forall i \in \{1, \ldots, q\}, \\ & x_e \geq 0 & \forall e \in E, \end{array} \qquad \begin{array}{lll} \max & \displaystyle\sum_{i=1}^{q} y_i & \\ \text{s.t.} & \displaystyle\sum_{i: e \in T_i} y_i \leq w_e & \forall e \in E, \\ & y_i \geq 0 & \forall i \in \{1, \ldots, q\}. \end{array}$$

The algorithm starts with the feasible dual solution $y = 0$ and the nonfeasible primal solution $x = \emptyset$. It iteratively increases the primal and dual solutions until the primal solution becomes feasible. In each iteration, if x is not feasible, then there exists a set T_k such that $x \cap T_k = \emptyset$. Such a subset is called *violated*. Indeed, the increase of the dual solution involves some dual variables corresponding to violated sets. Specifically, the increase of the dual variables depends on a *violation oracle* (called VIOLATION). In each iteration the violation oracle supplies a collection of violated subsets $\mathcal{V}_j \subseteq \{T_1, \ldots, T_q\}$, and the dual variables that correspond to subsets in $\mathcal{V}_j$ are increased *simultaneously and at the same speed.*[2] When x becomes feasible a *reverse delete step* is performed. This step removes as many elements as possible from the primal solution x as long as x remains feasible.

Let x^f denote the set output by the algorithm, and let ϵ_j denote the increase of the dual variables corresponding to $\mathcal{V}_j$. Thus, $y_i = \sum_{j:T_i \in \mathcal{V}_j} \epsilon_j$, $\sum_{i=1}^q y_i = \sum_{j=1}^\ell |\mathcal{V}_j| \, \epsilon_j$, and

$$\begin{aligned} w(x^f) &= \sum_{e \in x^f} w_e \\ &= \sum_{e \in x^f} \sum_{i:e \in T_i} y_i \\ &= \sum_{i=1}^q \left| x^f \cap T_i \right| y_i \\ &= \sum_{i=1}^q \left| x^f \cap T_i \right| \sum_{j:T_i \in \mathcal{V}_j} \epsilon_j \\ &= \sum_{j=1}^\ell \left(\sum_{T_i \in \mathcal{V}_j} \left| x^f \cap T_i \right| \right) \epsilon_j . \end{aligned}$$

Therefore, the weight of x^f is at most r times the value of the dual solution y (and, therefore, x^f is r-approximate) if for all $j \in \{1, \ldots, \ell\}$

$$\sum_{T_i \in \mathcal{V}_j} \left| x^f \cap T_i \right| \leq r \cdot \left| \mathcal{V}_j \right| . \tag{4.1}$$

Examine iteration j of the reverse deletion step. We know that when e_j was considered for removal, no element $e_{j'}$ with $j' < j$ had already been removed. Thus, after e_j is considered for removal, the temporary solution is $x^j = x^f \cup \{e_1, \ldots, e_{j-1}\}$. Observe that x^j is feasible and $x^j \setminus \{e\}$ is not feasible for all $e \in x^j \setminus \{e_1, \ldots, e_{j-1}\}$. x^j is called a *minimal augmentation* of $\{e_1, \ldots, e_{j-1}\}$ in [27]. Moreover,

$$\sum_{T_i \in \mathcal{V}_j} \left| x^f \cap T_i \right| \leq \sum_{T_i \in \mathcal{V}_j} \left| x^j \cap T_i \right| \ .$$

Thus, to obtain bound (4.1) Goemans and Williamson [27] set the following requirement on every collection of subsets $\mathcal{V}_j$: $\sum_{T_i \in \mathcal{V}_j} |x \cap T_i| \leq r \cdot |\mathcal{V}_j|$ for any minimal augmentation x of $\{e_1, \ldots, e_{j-1}\}$.

To summarize, in order to construct an r-approximate solution, in each iteration of the algorithm, we seek a collection $\mathcal{V}$ such that $\sum_{T_i \in \mathcal{V}} |x \cap T_i| \leq r \cdot |\mathcal{V}|$ for any minimal augmentation x of the current (nonfeasible) primal solution denoted by z.

[2] Some subsets in $\mathcal{V}_j$ may not be violated. See [27] for more details.

In essence we seek a collection $\mathcal{V}$ that satisfies a sort of amortized relaxed version of the dual complementary slackness conditions. We now formalize this demand from a collection of violated subsets in our terminology.

DEFINITION 4.1. *A collection $\mathcal{V} \subseteq \{T_1, \ldots, T_q\}$ is called* r-effective *with respect to $(\mathcal{F}, w, z)$ if $\sum_{T_i \in \mathcal{V}} |x \cap T_i| \leq r \cdot |\mathcal{V}|$ for any minimal feasible solution x with respect to $(\mathcal{F}, z)$.*

As did Bertsimas and Teo [14] we prefer to speak in terms of inequalities. An inequality is referred to as *valid* if any feasible solution to the problem at hand satisfies this inequality. For example, given an integer programming formulation of a problem, any inequality that appears in this formulation is valid. The following definition uses terms of inequalities and extends the previous definition.

DEFINITION 4.2. *A set of valid inequalities $\{\alpha^1 x \geq \beta^1, \ldots, \alpha^k x \geq \beta^k\}$ is called* r-effective *with respect to $(\mathcal{F}, w, z)$ if $\alpha^k_j = 0$ for every k and $j \in z$, and any integral minimal feasible solution x with respect to $(\mathcal{F}, z)$ satisfies $\sum_{i=1}^k \alpha^i x \leq r \cdot \sum_{i=1}^k \beta^i$.*

If this is true for any *integral feasible solution, the set is called* fully r-effective.

If an r-effective set contains a single inequality, we refer to this inequality as r-effective.

We remark that we require $\alpha^k_j = 0$ for every k and every $j \in z$ since in general we discuss inequalities with respect to $(\mathcal{F}, z)$ and not with respect to $\mathcal{F}$. If $z = \emptyset$, we sometimes say that the set (or the inequality) is r-effective with respect to $\mathcal{F}$.

An r-effective collection $\mathcal{V}$ can be understood as the r-effective set of valid inequalities $\{\sum_{e \in T_i} x_e \geq 1 : T_i \in \mathcal{V}\}$. However, Definition 4.1 allows the use of other kinds of inequalities, and therefore extends Definition 4.2. Thus, it would seem that our goal is to find an r-effective set of valid inequalities in each iteration. However, we show that it is enough to construct a *single* r-effective valid inequality for that purpose. Consider an r-effective set $S = \{\alpha^1 x \geq \beta^1, \ldots, \alpha^k x \geq \beta^k\}$ and the inequality that we get by summing up the inequalities in S:

$$\sum_{i=1}^{k} \alpha^i x = \sum_{j=1}^{n} \left(\sum_{i=1}^{k} \alpha^i \right)_j x_j \geq \sum_{i=1}^{k} \beta^i .$$

Since S is r-effective we know that $\sum_{i=1}^k \alpha^i x \leq r \cdot \sum_{i=1}^k \beta^i$, and we have found our r-effective inequality. Thus, our goal, in each iteration of the algorithm, is to find an inequality $\alpha x \geq \beta$ such that any minimal solution satisfies the following relaxed dual condition:

$$y_i > 0 \implies \alpha \cdot x \leq r\beta \ .$$

For example, examine the 2-approximation algorithm for the Steiner tree problem (Algorithm **PD-ST** of section 3). The r-effective collection of sets that is chosen by the algorithm in the ℓth iteration is $\mathcal{V} = \{(C, \bar{C}) : C \in \mathcal{C}'_\ell\}$. The corresponding r-effective collection of valid inequalities is $S = \{\sum_{e \in (C, \bar{C})} x_e \geq 1 : C \in \mathcal{C}'_\ell\}$. Consider the inequality that we get by summing up the inequalities in S:

$$\sum_{C \in \mathcal{C}'_\ell} \sum_{e \in (C, \bar{C})} x_e = \sum_{e \in E^\ell} \tau_{G^\ell}(e) x_e \geq |\mathcal{C}'_\ell| \,, \tag{4.2}$$

where E^ℓ is the edge set of G^ℓ. Clearly, (4.2) is valid and, by Lemma 3.1, is also 2-effective. Notice that the coefficients of (4.2) and the weights that are used in Algorithm **LR-ST** are identical. As we shall see in what follows, this is no coincidence.

```
Algorithm PDcov($z, w, k$).
1.  If $\emptyset \in \textsc{Sol}(\mathcal{F}, z)$ return $\emptyset$
2.  Construct a valid inequality $\alpha^k x \geq \beta^k$
        which is $r$-effective w.r.t. $(\mathcal{F}, z)$
3.  $y_k \leftarrow \max\left\{\epsilon : w - \epsilon\alpha^k \geq 0\right\}$
4.  Let $j \notin z$ be an index for which $w_j = y_k \alpha_j^k$
5.  $x \leftarrow \mathbf{PDcov}(z \cup \{j\}, w - y_k\alpha^k, k+1)$
6.  If $x \notin \textsc{Sol}(\mathcal{F}, z)$ then $x \leftarrow x \cup \{j\}$
7.  Return $x$
```

Fig. 4.2.

Bertsimas and Teo [14] proposed a generic algorithm to design and analyze primal-dual approximation algorithms for problems of the following type:

$$\begin{array}{ll} \min & wx \\ \text{s.t.} & Ax \geq b, \\ & x \in \{0,1\}^n, \end{array}$$

where A, b, and w are nonnegative. This algorithm constructs a single valid inequality in each iteration and uses it to modify the current instance. The size of the problem instance is reduced in each iteration, and therefore the algorithm terminates after no more than n iterations. The approximation ratio of this algorithm depends on the choice of the inequalities. In fact, it corresponds to what Bertsimas and Teo call the *strength* of the inequalities. In our terminology, the *strength* of an inequality is the minimal value of r for which it is r-effective. It is important to note that, unlike other primal-dual algorithms, this algorithm constructs new valid inequalities during its execution. Another difference is that it uses the weight vector in order to measure the tightness of the dual constraints. Thus, in each iteration it decreases the weights according to the inequality that was used. In fact, this study was inspired by the similarity between this weight decrease and its local ratio counterpart.

Algorithm **PDcov** is a recursive version of the algorithm from [14]; see Figure 4.2. The initial call is $\mathbf{PDcov}(\emptyset, w, 1)$. (The third parameter is used for purposes of analysis.) Informally, it can be viewed as follows: construct an r-effective inequality; update the corresponding dual variable and w such that w remains nonnegative; find an element j whose weight is zero; add j to the temporary partial solution z; then recursively solve the problem with respect to $\mathcal{F}, z$ and the new weights (the termination condition of the recursion is met when the empty set becomes feasible); finally, j is added to the solution x only if it is necessary.

The following analysis is based on the corresponding analysis from [14].

We start by proving by induction on the recursion that Algorithm **PDcov** returns minimal feasible solutions with respect to (F, z). At the recursion basis the solution returned is the empty set, which is both feasible and minimal. For the inductive step, let x' be the solution returned by the recursive call in line 5. x' is feasible with respect to $(\mathcal{F}, z \cup \{j\})$ by the inductive hypothesis; therefore x is feasible with respect to (F, z). We show that $x \setminus \{i\}$ is not feasible for every $i \in x$. For the case where $i \neq j$, if $x \setminus \{i\}$ is feasible with respect to $(\mathcal{F}, z)$, then $x' \setminus \{i\}$ is feasible with respect to $(\mathcal{F}, z \cup \{j\})$ in contradiction with the minimality of x'. The case where $i = j$, which is relevant only when $x = x' \cup \{j\}$, is trivial.

Next we show that the algorithm returns r-approximate solutions. Consider the following linear program

$$\text{(P)} \qquad \begin{array}{ll} \min & wx \\ \text{s.t.} & \alpha^k x \geq \beta^k \quad k \in \{1, \dots, t\}, \\ & x \geq 0, \end{array}$$

where $\alpha^k x \geq \beta^k$ is the inequality used in the kth recursive call, and $t+1$ is the recursion depth. The dual is

$$\text{(D)} \qquad \begin{array}{ll} \max & \beta y \\ \text{s.t.} & \displaystyle\sum_{k=1}^{t} \alpha_j^k y_k \leq w_j \quad j \in \{1, \dots, n\}, \\ & y \geq 0. \end{array}$$

Examine the kth recursive call. Let z^k be the temporary partial solution at depth k. $\alpha^k x \geq \beta^k$ is a valid inequality with respect to $(\mathcal{F}, z^k)$, and, therefore, it is valid with respect to $\mathcal{F}$. Thus, $\textsc{Sol}(\mathcal{F}) \subseteq \textsc{Sol}(\text{P})$, and $\textsc{Opt}(\text{P}) \leq \textsc{Opt}(\mathcal{F}, w)$. As we have seen before, x is a feasible solution for $\mathcal{F}$ and, therefore, for (P). Also, y is a feasible solution for the dual of (P).

Let x^k be the solution returned by the kth recursive call. Also, let w^k be the weight vector, and let j be the chosen element at the kth call. We prove by induction that $w^k x^k \leq r \sum_{l \geq k} y_l \beta^l$. First, for $k = t+1$, we have $w^{t+1} x^{t+1} = 0 = \sum_{l \geq t+1} y_l \beta^l$. For $k \leq t$ we have

$$\begin{aligned} w^k x^k &= (w^{k+1} + y_k \alpha^k) x^k \\ &= w^{k+1} x^{k+1} + y_k \alpha^k x^k && (4.3) \\ &\leq r \sum_{l \geq k+1} y_l \beta^l + y_k r \beta^k && (4.4) \\ &= r \sum_{l \geq k} y_l \beta^l, \end{aligned}$$

where (4.3) is due to the fact that $w_j^{k+1} = 0$, and (4.4) is implied by the induction hypothesis and the r-effectiveness of the inequality $\alpha^k x \geq \beta^k$. Finally, x is r-approximate since

$$wx = w^1 x^1 \leq r \sum_{l \geq 1} y_l \beta^l \leq r \cdot \textsc{Opt}(\text{P}) \leq r \cdot \textsc{Opt}(\mathcal{F}, w) .$$

We remark that the value of y_k depends on the coefficients of the valid inequality $\alpha^k x \geq \beta$. That is, we can use the valid inequality $\rho \cdot \alpha^k x \geq \rho \cdot \beta$ for any $\rho > 0$ instead of using $\alpha^k x \geq \beta$, provided that the value of y_k is divided by ρ. In fact, by choosing the appropriate value of ρ, we can always ensure that $y_k = 1$. This fact is used in what follows.

4.2. A local ratio framework for covering problems. As was demonstrated in section 3 the typical step of a local ratio algorithm involves the construction of a "good" weight function. Algorithm **LR-ST** used a weight function such that any minimal Steiner tree is 2-approximate with respect to it. In [4] Bar-Yehuda defined this notion of goodness in the context of covering. The definition is given in our terminology.

```
Algorithm LRcov($z, w$).
    If $\emptyset \in \textsc{Sol}(\mathcal{F}, z)$ return $\emptyset$
    Construct a $w$-tight weight function $\delta$
            which is $r$-effective w.r.t. $(\mathcal{F}, z)$
    Let $j \notin z$ be an index for which $\delta_j = w_j$
    $x \leftarrow$ LRcov$(z \cup \{j\}, w - \delta)$
    If $x \notin \textsc{Sol}(\mathcal{F}, z)$ then $x \leftarrow x \cup \{j\}$
    Return $x$
```

Fig. 4.3.

Definition 4.3 (see [4]). *Given a covering problem $(\mathcal{F}, w, z)$, a weight function δ is called* r-effective *with respect to $(\mathcal{F}, z)$ if for all $j \in z, \delta_j = 0$, and if every minimal feasible solution x with respect to $(\mathcal{F}, z)$ satisfies $\delta x \leq r \cdot \textsc{Opt}(\mathcal{F}, \delta, z)$.*

We prefer the following equivalent (yet more practical) definition.

Definition 4.4. *Given a covering problem $(\mathcal{F}, w, z)$, a weight function δ is called* r-effective *with respect to $(\mathcal{F}, z)$ if for all $j \in z, \delta_j = 0$, and if there exists β such that every minimal feasible solution x with respect to $(\mathcal{F}, z)$ satisfies $\beta \leq \delta \cdot x \leq r\beta$. In this case we say that β is a* witness *to δ's r-effectiveness.*

If this is true for any *integral feasible solution δ is called* fully r-effective.

We remark that we require $\delta_j = 0$ for every $j \in z$ since in general we deal with inequalities with respect to $(\mathcal{F}, z)$ and not with respect to $\mathcal{F}$. If $z = \emptyset$, we say that δ is r-effective with respect to $\mathcal{F}$.

Obviously, by assigning $\beta = \delta x^*$, where x^* is an optimal solution, we get that the first definition implies the latter. For the other direction, notice that $\beta \leq \delta x^*$.

A local ratio algorithm for a covering problem works as follows. First, construct an r-effective weight function δ such that $\delta \leq w$ and there exists some j for which $w_j = \delta_j$. Such a weight function is called *w-tight*. Subtract δ from the weight function w. Add all zero-weight elements to the partial solution z. Then recursively solve the problem with respect to $(\mathcal{F}, w - \delta, z)$. When the empty set becomes feasible (or when z becomes feasible with respect to $\mathcal{F}$) the recursion terminates. Finally, remove unnecessary elements from the temporary solution by performing a reverse deletion phase.

Algorithm **LRcov** is a generic approximation algorithm for covering problems; see Figure 4.3. (The initial call is **LRcov**$(\emptyset, w)$.) The main difference between the algorithm from [4] and the one given here is that in the latter the augmentation of the temporary solution is done one element at a time. By doing this we have the option not to include zero-weight elements which do not contribute to the feasibility of the partial solution z. When using the algorithm from [4] such elements are removed during the reverse deletion phase (called *removal loop* in [4]). In order to simulate the algorithm from [4], when using Algorithm **LRcov** we can add zero weight elements one by one. This is due to the fact that $\delta = 0$ is r-effective for all $r \geq 1$.

Proving that Algorithm **LRcov** returns minimal feasible solutions with respect to $(\mathcal{F}, z)$ is essentially identical to proving that Algorithm **PDcov** returns minimal feasible solutions (see section 4.1). Thus, we need only to prove that Algorithm **LRcov** outputs an r-approximate solution.

We prove by induction on the recursion that Algorithm **LRcov** returns an r-

approximation with respect to $(\mathcal{F}, w, z)$. At the recursion basis, $\emptyset$ is an optimal solution. Otherwise, for the inductive step, examine x at the end of the recursive call. By the induction hypothesis $x \setminus \{j\}$ is an r-approximation with respect to $(\mathcal{F}, w - \delta, z \cup \{j\})$. Moreover, due to the fact that $w_j - \delta_j = 0$, x is r-approximate with respect to $(\mathcal{F}, w - \delta, z)$. Finally, by the r-effectiveness of δ and the local ratio theorem we get that x is an r-approximate solution with respect to $(\mathcal{F}, w, z)$ as well.

4.3. Equivalence. It is not hard to see that Algorithm **PDcov** and Algorithm **LRcov** share the same structure. Both algorithms, in each recursive call, modify the weights, add a zero-weight element to z, and solve the problem recursively. The only difference between the two is that Algorithm **PDcov** uses r-effective inequalities, while Algorithm **LRcov** constructs r-effective weight functions. The following lemma shows that an r-effective valid inequality and an r-effective weight function are one and the same.

LEMMA 4.5. *$\alpha x \geq \beta$ is an r-effective inequality if and only if α is an r-effective weight function with β as a witness.*

Proof. Let $\alpha x \geq \beta$ be an r-effective inequality. By definition every minimal feasible solution x satisfies $\beta \leq \alpha x \leq r\beta$. Thus, α is an r-effective weight function. On the other hand, let α be an r-effective weight function with a witness β. Due to the r-effectiveness of α every minimal feasible solution x satisfies $\beta \leq \alpha x \leq r\beta$. Therefore, $\alpha x \geq \beta$ is an r-effective inequality. □

We remark that when using an r-effective weight function δ, Algorithm **LRcov** does not need to know the value of the witness to δ's r-effectiveness. In fact, it can be NP-hard to calculate this value. The same goes for Algorithm **PDcov**. We do not have to know the value of the right-hand side of an r-effective inequality $\alpha x \geq \beta$. This is demonstrated in section 4.4.4.

By Lemma 4.5 the use of an inequality can be simulated by utilizing the corresponding weight function, and vice versa. Thus, the primal-dual schema and the local ratio technique converge on standard applications.

COROLLARY 4.6. *Algorithms **PDcov** and **LRcov** are identical. Moreover, the equivalence is constructive; i.e., any implementation of one can be transformed into an implementation of the other.*

Although both algorithms are equivalent, the analysis of Algorithm **PDcov** seems more complicated than the analysis of Algorithm **LRcov**. The difference is artificial. The local ratio technique uses a *local* approach. A typical local ratio advancement step is local in the sense that it can be analyzed independently of the rest of the algorithm (see also [4]). Therefore, local ratio algorithms tend to be recursive and their analyses inductive. On the other hand, primal-dual analyses use a more *global* approach. Instead of comparing intermediate weights, the total weight of the integral primal solution is compared to the cost of the dual solution. This approach is also used outside the primal-dual schema (e.g., [34, 33]). The equivalence implies that there is no need to use the global approach in the context of the primal-dual schema. Indeed, the analysis of Algorithm **PDcov** uses exactly the same local arguments as the analysis of Algorithm **LRcov**.

In the analysis of Algorithm **PDcov** we compared the integral primal solution x to a dual solution y in order to prove that the former is r-approximate. Recall that y was not a dual solution to the original program. We have defined a new program, called (P), that contains the valid inequalities that were used by the algorithm, and the primal solution was compared to the dual of (P). Clearly, the best approximation ratio we can hope for using this approach is the *integrality gap* of (P). Thus, one can check whether an analysis for an algorithm is tight by comparing the performance ratio given by the analysis to the integrality gap of (P). Now, consider the set of weight functions that were used by an implementation of Algorithm **LRcov**. The corresponding inequalities would be the constraints of (P). Thus, one can check whether an analysis of a local ratio algorithm is tight by calculating the integrality gap of (P) as well.

4.4. Applications. When trying to approximate a minimization problem we need to address several issues that depend on the combinatorial structure of the problem at hand. First and foremost, we need to construct valid r-effective inequalities, or r-effective weight functions. Also, we need to use them such that the algorithm terminates in polynomial time. The algorithms for covering problems make use of the fact that you can add a zero-weight element to the temporary partial solution and, by doing so, reduce the size of the problem. This ensures that the running time is polynomial. Also, this allows us to use inequalities or weight functions which are r-effective with respect to the current instance, but are not necessarily so with respect to the original instance. Many covering problems were approximated by making use of this mechanism (e.g., feedback vertex set [2] and network design problems [27]). This is demonstrated in what follows. Namely, we illustrate how Algorithms **PD-cov** and **LRcov** can be used to construct and analyze approximation algorithms for covering problem. Note that when an algorithm is presented it is not given in full detail. We only describe the valid inequalities or weight functions needed in order to implement it using one of the generic algorithms.

Many approximation algorithms for covering problems use only one type of inequality or weight function. Such algorithms rely on the fact that when an instance is modified (or when an element is added to z, in our terminology) the resulting instance is still an instance of the same covering problem. For example, when Algorithm **LR-ST** contracts an edge the resulting instance is still an instance of the *Steiner tree* problem. Bertsimas and Teo [14] call an integer programming formulation that satisfies this property *reducible*. Thus, in such cases, it is enough to describe and analyze an inequality or a weight function with respect to the original set of constraints $\mathcal{F}$.

4.4.1. Steiner tree and other network design problems. Let $\mathcal{F}$ be a set of constraints for the *Steiner tree* problem (e.g., the inequalities in (ST)). Consider the instance $(\mathcal{F}, z)$ for some vector z. Recall that the elements (i.e., edges) in z are assumed to be taken into the solution. Thus, an instance $(\mathcal{F}, z)$ contains components on which there are connectivity demands. Bearing this in mind it is not hard to see that Algorithm **LR-ST** (see Figure 3.2) is an implementation of Algorithm **LRcov**. In each recursive call the algorithm uses the weight function $\delta(e) = \epsilon \cdot \tau(e)$, where $\epsilon = \min_e \{w(e)/\tau(e)\}$, and then contracts a zero-weight edge. (Recall that $\tau(e)$ is the number of terminals incident to e.) This contraction can be represented by adding the edge e to z.

While Algorithm **LR-ST** can be viewed as an implementation of Algorithm **LRcov**, Algorithm **PD-ST** is not an implementation of Algorithm **PDcov**. For starters Algorithm **PD-ST** is iterative and not recursive. Also, it raises several dual variables in each iteration, and not one. However, as demonstrated in section 4.1, when summing up the inequalities that correspond to the dual variables that are raised in an iteration, we get (4.2), which is 2-effective. Therefore, it is enough to raise a single dual variable corresponding to (4.2) in each recursive call of Algorithm **PDcov**.

Algorithm **PD-ST** is a special case of an algorithm for *constrained forest* problems given by Goemans and Williamson [26]. Given a graph $G = (V, E)$, a function f :

$2^V \to \{0,1\}$, and a nonnegative weight function w on the edges, they have considered the integer program

$$\begin{array}{lll} \min & \sum_{e\in E} w_e x_e & \\ \text{s.t.} & \sum_{e\in\delta(S)} x_e \geq f(S) & \forall S,\ \emptyset \subsetneq S \subsetneq V, \\ & x_e \in \{0,1\} & \forall e \in E, \end{array}$$

where $\delta(S)$ denotes the set of edges having exactly one endpoint in S. They presented a $(2-2/|A|)$-approximation algorithm, where $A = \{v : f(v) = 1\}$, for the case where f is *proper*.[3] In [27] Goemans and Williamson showed that the same algorithm outputs a 2-approximate solution in the case of *downwards monotone* functions.[4] Williamson et al. [40] generalized this algorithm for the class of *uncrossable* functions.[5] They used this generalization to present a multiphase primal-dual $2f_{\max}$-approximation algorithm for general proper functions, where $f_{\max} = \max_S f(S)$. They reduced the problem to a sequence of hitting set problems and applied the primal-dual approximation algorithm for *uncrossable* functions to each subproblem. Thus, the solution to the original problem is the union of the solutions of the subproblems. Consequently, Goemans et al. [25] improved the approximation ratio to $2\mathcal{H}(f_{\max})$, where $\mathcal{H}$ is the harmonic function. (For more details see [27].)

Bertsimas and Teo [14] showed that (4.2) is 2-effective even when f is *uncrossable*. Thus, all the above algorithms can be implemented using Algorithm **PDcov**. Moreover, because τ is a 2-effective weight function, all of them can be explained by local ratio means using Algorithm **LRcov**. In fact, the multiphase primal-dual algorithms from [40, 25] can be analyzed as multiphase local ratio algorithms. In [5] Bar-Yehuda et al. presented the algorithm from [26] in local ratio terms and, in particular, showed that τ is 2-effective for *proper* and *downwards monotone* functions.

4.4.2. Generalized hitting set. The *generalized hitting set* problem is defined as follows. Given a collection of subsets S of a ground set E, a nonnegative weight $w(s)$ for every set $s \in S$, and a nonnegative weight $w(u)$ for every element $u \in E$, find a minimum-weight collection of objects $C \subseteq E \cup S$, such that for all $s \in S$, either there exists $u \in C$ such that $u \in s$ or $s \in C$. As in the *hitting set* problem our objective is to hit all the sets in S by using elements from E. However, in this case, we are allowed not to cover a set s, provided that we pay a tax $w(s)$. The hitting set problem is the special case where the tax is infinite for all sets. The generalized hitting set problem can be formalized as follows:

$$\begin{array}{lll} \min & \sum_{u\in E} w(u)x_u + \sum_{s\in S} w(s)x_s & \\ \text{s.t.} & \sum_{u\in s} x_u + x_s \geq 1 & \forall s \in S, \\ & x_t \in \{0,1\} & \forall t \in E \cup S, \end{array}$$

where $x_u = 1$ if and only if u is in the cover, and $x_s = 1$ if and only if s is not hit.

[3] A function f is *proper* if (1) for all $S \subsetneq V$, $f(S) = f(V \setminus S)$; and (2) for all S, T, $S \cap T = \emptyset$, $f(S \cup T) \leq \max\{f(S), f(T)\}$.

[4] A function f is *downwards monotone* if $f(S) = 1$ implies $f(S') = 1$ for all nonempty $S' \subseteq S$.

[5] A function f is *uncrossable* if (1) for all $S \subsetneq V$, $f(S) = f(V \setminus S)$, and (2) if S, T are intersecting sets such that $f(S) = f(T) = 1$, then either $f(S \setminus T) = f(T \setminus S) = 1$ or $f(S \cap T) = f(S \cup T) = 1$.

Observe that paying the tax $w(s)$ is required only when s is not hit. Thus, the inequality $\sum_{u\in s} x_u + x_s \geq 1$ is a Δ-effective inequality for any set $s \in S$, where $\Delta = \max\{|s| : s \in S\}$. The corresponding Δ-effective weight function is

$$\delta(t) = \begin{cases} \epsilon, & t \in \{s\} \cup s, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, a Δ-approximation algorithm can be constructed using one of the frameworks. We remark that the above inequalities remain Δ-effective if we use any value between 1 and Δ as x_s's coefficient. Analogously, any value between ϵ and $\Delta \cdot \epsilon$ is acceptable for $\delta(s)$.

A linear time Δ-approximation algorithm can be obtained by extending the Δ-approximation algorithm for hitting set [6]. Use the above inequalities (weight functions) in an arbitrary order; then construct a zero-weight minimal feasible solution as follows: pick all zero-weight elements and all the sets which are not hit by some zero-weight element. When $\Delta = 2$ we get a special case called *generalized vertex cover*, for which Hochbaum [31] presented an $\mathrm{O}(nm \log n^2/m)$ 2-approximation algorithm.

4.4.3. Feedback vertex set in tournaments. A *tournament* is an orientation of a complete (undirected) graph; i.e., it is a directed graph with the property that for every unordered pair of distinct vertices $\{u, v\}$ it either contains the arc (u, v) or the arc (v, u), but not both. The *feedback vertex set in tournaments* problem is the following. Given a tournament and a weight function w on its vertices, find a minimum-weight set of vertices whose removal leaves a graph containing no directed cycles.

It is not hard to verify that a tournament contains a directed cycle if and only if it contains a *triangle*, where a triangle is a directed cycle of length 3. Thus, we may restrict our attention to triangles and formulate the problem as follows:

$$\text{(FVST)} \qquad \begin{array}{ll} \min & \sum_{v\in V} w_v x_v \\ \text{s.t.} & \sum_{v\in T} x_v \geq 1 \quad \forall \text{ triangle } T, \\ & x_v \in \{0,1\} \quad \forall v \in V. \end{array}$$

We say that a triangle is *positive* if all of its vertices have strictly positive weights. Clearly, the set of all zero-weight vertices is an optimal solution (of zero-weight) if and only if the tournament contains no positive triangles. Thus we obtain a 3-approximation algorithm by means of the following 3-effective weight function. Let $\{v_1, v_2, v_3\}$ be a positive triangle and let $\epsilon = \min\{w(v_1), w(v_2), w(v_3)\}$. Define

$$\delta(v) = \begin{cases} \epsilon, & v \in \{v_1, v_2, v_3\}, \\ 0 & \text{otherwise.} \end{cases}$$

The maximum cost, with respect to δ, of a feasible solution is clearly at most 3ϵ, while the minimum cost is at least ϵ, since every feasible solution must contain at least one of v_1, v_2, v_3. The corresponding 3-effective inequality is $x_{v_1} + x_{v_2} + x_{v_3} \geq 1$.

Note that *any* feasible solution is 3-approximate with respect to δ (not only minimal solutions). Equivalently, the inequality $x_{v_1} + x_{v_2} + x_{v_3} \leq 3$ holds for *any* feasible solution. Thus, the weight function and inequality are fully r-effective.

4.4.4. Feedback vertex set. A set of vertices in an undirected graph is called a *feedback vertex set* if its removal leaves an acyclic graph (i.e., a forest). In other words, the set must cover all cycles in the graph. The *feedback vertex set* problem is as follows: given a vertex-weighted graph, find a minimum weight feedback vertex set.

Bafna, Berman, and Fujito [2] presented a local ratio 2-approximation algorithm for the *feedback vertex set* problem. Their algorithm can be implemented using Algorithm **LRcov**. A cycle C is *semidisjoint* if there exists $x \in C$ such that $\deg(u) = 2$ for every vertex $u \in C \setminus \{x\}$. If G contains a semidisjoint cycle C, let $\epsilon = \min_{v \in C} w(v)$, and use the 1-effective weight function

$$\delta_1(v) = \begin{cases} \epsilon, & v \in C, \\ 0 & \text{otherwise;} \end{cases}$$

otherwise use the weight function $\delta_2(v) = \epsilon \cdot (\deg(v) - 1)$, where $\epsilon = \min_{v \in V} \{w(v)/(\deg(v) - 1)\}$. Bafna, Berman, and Fujito [2] showed that δ_2 is 2-effective in graphs that (1) do not contain semidisjoint cycles, and (2) $\deg(v) \geq 2$ for every $v \in V$. In order to implement this algorithm using Algorithm **PDcov** one should use the following valid inequalities: $\sum_{v \in C} x_v \geq 1$ in case G contains a semidisjoint cycle C, and $\sum_{v \in V} (\deg(v) - 1) \cdot x_v \geq |E| - |V| + 1$ otherwise.

Another 2-approximation algorithm is due to Becker and Geiger [13]. In [4] Bar-Yehuda indicated that their algorithm can be restated in local ratio terms with the weight function $\delta(v) = \deg(v)$, which is 2-effective. It can be shown that the corresponding 2-effective inequality is $\sum_{v \in V} \deg(v) x_v \geq |E| - |V| + 1 + \tau$, where τ is the cardinality of the smallest feedback vertex set in G. Therefore, a primal-dual analysis to this algorithm can be given by using Algorithm **PDcov**. It is important to note that we do not need to know the value τ in order to execute the algorithm. In fact, this value is NP-hard to compute.

Chudak et al. [17] explained both algorithms using primal-dual and added a third 2-approximation algorithm which is similar to the one from [2]. We present it as an implementation of Algorithm **PDcov**. That is, we show which inequality to use in each recursive call. An *end-block* is a biconnected component containing at most one articulation point. Choose an end-block B and use the inequality $\sum_{v \in V} (\deg(v) - 1) x_v \geq |E| - |V| + 1$. The corresponding weight function is

$$\delta(v) = \begin{cases} \epsilon \cdot (\deg(v) - 1), & v \in B, \\ 0 & \text{otherwise.} \end{cases}$$

Local ratio implementations of the three algorithms and a detailed analysis of the one from [13] can be found in [5].

5. Minimization frameworks.

The recursive algorithms for covering problems can be divided into three primitives: the recursion base, the way that an instance is modified before a recursive call, and the way in which the solution returned by a recursive call is fixed. In this section we present a more general framework that can explain many algorithms that do not fall within the scope of our generic algorithms for covering. This is done by means of extending each of the three primitives mentioned above.

Modifying the instance. The frameworks for covering problems rely heavily on the fact that the set of constraints $\mathcal{F}$ is monotone. In each recursive call the current instance is modified by assuming that a zero-weight element is taken into the solution (i.e., by adding a zero-weight element to z). This can be done because in covering problems adding a zero-weight element to the solution is never a bad move. However, in the noncovering case, a solution containing this element may not even exist. Also, in nonboolean problems, there are several possible assignments for a zero-weight variable. Thus, we need to extend the algorithms by considering more ways in which to modify the instance.

Fixing solutions. After each recursive call the covering algorithms fix, if necessary, the solution returned in order to turn it into a "good" solution, i.e., into a minimal solution. This is done because the algorithms use weight functions or inequalities that are r-effective. The solution returned by the recursive call is fixed in a very straightforward manner—add the element that was removed from the instance to the solution if it is not feasible. It turns out that an algorithm may use weight functions or inequalities for which good solutions are solutions that satisfy a certain property different from *minimality.* In fact, this property can be simply *the solutions returned by the algorithm.* We refer to such weight functions and inequalities as *r-effective with respect to a property $\mathcal{P}$.* Clearly, in such cases, the algorithm may be forced to fix the solution returned by a recursive call in a way that is very different from simply adding a single element in case the current solution is not feasible.

Recursion base. By adding a new element to z in each recursive call of Algorithm **LRcov** (or **PDcov**), we are bound to arrive at the recursion base, which is the empty instance, and for which the empty set is always a minimal optimal solution. However, other recursion bases are possible. In [7] Bar-Yehuda and Even developed a $(2 - \frac{\log\log n}{2\log n})$-approximation algorithm for a *vertex cover* which is partly based on local ratio. Their algorithm starts with a local ratio phase that removes short odd cycles from the graph, and then continues to the next phase that finds approximate solutions for graphs that do not have short odd cycles. This can be explained by a variant of Algorithm **LRcov** in which the recursion base is replaced by the invocation of an approximation algorithm that works only for inputs of a certain kind and returns r-approximate minimal solutions. (The solution need not be minimal if the weight functions used are fully r-effective.)

5.1. The algorithms. Our framework can be described as follows. In each recursive call the algorithm constructs and uses a weight function or an inequality and modifies the instance. Then it recursively solves the problem on the new instance and the new objective function. Afterwards, it fixes the solution returned. The recursion base is performed if an instance satisfies some property $\mathcal{Q}$.

We use the following three subroutines:

- **Modify**$(\mathcal{F}, w)$: Modifies the current instance by assigning values to zero-weight variables and then removing them. This subroutine modifies an instance such that any valid inequality with respect to the modified instance is also valid with respect to the current instance (and hence to the original instance as well).
- **Fix**$(\mathcal{F}, w, x', \mathcal{P})$: Given an r-approximate solution x' for the instance **Modify**$(\mathcal{F}, w)$, returns an r-approximate solution x for the instance $(\mathcal{F}, w)$ satisfying some property $\mathcal{P}$. The solution x is constructed from x' by changing only zero-weight variables. Note that each recursive call may use a different property.
- **Base**$(\mathcal{F}, w)$: Given a problem instance that satisfies $\mathcal{Q}$ returns an r-approximate solution.

```
Algorithm LRmin($\mathcal{F}, w$).
    1. If $\mathcal{F}$ satisfies $\mathcal{Q}$ return Base($\mathcal{F}, w$)
    2. Construct a weight function $\delta$ which is $r$-effective with respect to
       a property $\mathcal{P}$ such that $w - \delta \geq 0$
    3. $\mathcal{F}' \leftarrow$ Modify($\mathcal{F}, w - \delta$)
    4. $x' \leftarrow$ LRmin($\mathcal{F}', w - \delta$)
    5. $x \leftarrow$ Fix($\mathcal{F}, w - \delta, x', \mathcal{P}$)
    6. Return $x$
```

Fig. 5.1.

```
Algorithm PDmin($\mathcal{F}, w$).
    1. If $\mathcal{F}$ satisfies $\mathcal{Q}$ return Base($\mathcal{F}, w$)
    2. Construct an inequality $\alpha x \geq \beta$ which is $r$-effective with respect to
       a property $\mathcal{P}$ such that $w - \alpha \geq 0$
    3. $\mathcal{F}' \leftarrow$ Modify($\mathcal{F}, w - \alpha$)
    4. $x' \leftarrow$ PDmin($\mathcal{F}', w - \alpha$)
    5. $x \leftarrow$ Fix($\mathcal{F}, w - \delta, x', \mathcal{P}$)
    6. Return $x$
```

Fig. 5.2.

This time we start with the local ratio algorithm.

The analysis of Algorithm **LRmin** (see Figure 5.1) is similar to the analysis of Algorithm **LRcov**. We prove that the algorithm returns an r-approximate solution by induction on the recursion. The recursion base is trivial since subroutine **Base** returns r-approximate solutions by definition. For the inductive step, consider the solution x' that was returned by the recursive call. By the inductive hypothesis x' is r-approximate with respect to $(\mathcal{F}', w - \delta)$. Due to subroutines **Modify** and **Fix** x is r-approximate with respect to $(\mathcal{F}, w - \delta)$ and satisfies property $\mathcal{P}$. Furthermore, δ is r-effective with respect to $\mathcal{P}$. Thus, by the local ratio theorem x is also r-approximate with respect to $(\mathcal{F}, w)$.

Algorithm **PDmin** (see Figure 5.2) is our primal-dual approximation algorithm. It uses the same three primitives that are used by Algorithm **LRmin**.

We show that Algorithm **PDmin** returns r-approximate solutions. We do that by generalizing the analysis of Algorithm **PDcov**. Let $t + 1$ be the recursion depth. Let $(\mathcal{F}_k, w^k)$ denote the instance given to the kth recursive call, and let x^k denote the solution returned by the kth recursive call. Consider the linear program

$$\text{(P)} \qquad \begin{array}{lll} \min & wx & \\ \text{s.t.} & \alpha^k x \geq \beta^k & k \in \{1, \ldots, t+1\}, \\ & x \geq 0, & \end{array}$$

where $\alpha^k x \geq \beta^k$ for $k \in \{1, \ldots, t\}$ is the inequality used in the kth recursive call, $\alpha^{t+1} = w^{t+1}$, and $\beta^{t+1} = w^{t+1} \cdot x^{t+1}/r$. Due to subroutine **Modify**, and since $\alpha^{t+1} x \geq \beta^{t+1}$ for every solution x, (P) is a relaxation of $\mathcal{F}$, and therefore $\text{Opt(P)} \leq \text{Opt}(\mathcal{F}, w)$.

Let us build a solution y to the dual of (P), that is denoted by (D). Let (P_k) be the linear program that we get from (P) by discarding the first $k-1$ inequalities and changing the objective function to $w^k x$, and let (D_k) be the dual of (P_k). Consider the base instance $(\mathcal{F}_{t+1}, w^{t+1})$. Subroutine **Base** returns a solution x^{t+1} whose weight is no more than r times the optimal solution of $(\mathcal{F}_{t+1}, w^{t+1})$. x is also r-approximate with respect to (P_{t+1}). (Note that (P_{t+1}) contains only one constraint.) Thus, $w^{t+1}x^{t+1}$ is bounded by r times the value of $y^* = 1$ which is an optimal solution to (D_{t+1}). Let y be a vector of size $t+1$ whose entries are all 1. Let y^k be the vector that consists of $t-k+1$ 1's. That is, y^k is a vector that contains the last $t-k+1$ entries of y. We prove by induction that y^k is a solution to (D_k) for all k, which implies that y is a feasible solution of (D) (since $y = y^1$, and (D) $= (\mathrm{D}_1)$). At the base of the recursion, $y^{t+1} = y^*$ is an optimal solution to (D_{t+1}). For the inductive step, we assume that y^{k+1} is a solution to (D_{k+1}) and prove that y^k is a solution to (D_k). First, we claim that $(0, y^{k+1})$ (a vector consisting of a zero followed by the entries of y^{k+1}) is a feasible solution to (D_k). To see this, notice that a packing of constraints from (P_{k+1}) is also a packing of constraints from (P_k). Thus, $y^k = (1, y^{k+1})$ is also a packing of constraints from (P_k), since $w^k = w^{k+1} + \alpha^k$.

We can now analyze the approximation ratio. We prove by induction that $w^k x^k \leq r \sum_{l \geq k} y_l \beta^l$ for all k. For $k = t+1$, this is true since $\beta^{t+1} = \alpha^{t+1}x^{t+1}/r$. For $k \leq t$ we have

$$\begin{aligned} w^k x^k &= (w^{k+1} + \alpha^k)x^k \\ &= w^{k+1}x^{k+1} + y_k \alpha^k x^k \qquad (5.1) \\ &\leq r \sum_{l \geq k+1} y_l \beta^l + y_k r \beta^k \qquad (5.2) \\ &= r \sum_{l \geq k} y_l \beta^l, \end{aligned}$$

where (5.1) stems from the fact that subroutine **Fix** changes only zero-weight variables, and (5.2) is due to the induction hypothesis, the fact that subroutine **Fix** returns solutions with property $\mathcal{P}$, and the r-effectiveness of the inequality $\alpha^k x \geq \beta^k$ with respect to $\mathcal{P}$. x is r-approximate since

$$wx = w^1 x^1 \leq r \sum_{l \geq 1} y_l \beta^l \leq r \cdot \mathrm{O\small{PT}}(\mathrm{P}) \leq r \cdot \mathrm{O\small{PT}}(\mathcal{F}, w) \ .$$

5.2. Discussion. The only varying elements in the framework for covering are the r-effective inequalities (weight functions). That is, in order to construct an algorithm for a covering problem one has to find the appropriate inequalities (weight functions), and the rest is determined by the framework. The task of designing an algorithm may be much more complicated when one chooses to use the framework given in this section. For starters one has to come up with a suitable and polynomial implementation of subroutines **Base**, **Modify**, and **Fix**. Also, the resulting algorithm must reach the recursion base in polynomial time. Intuitively, after finding an r-effective inequality (weight function), we must ask ourselves the following question: How should we remove zero-weight elements? We must be able to remove zero-weight elements in a way that enables us to later fix the solution returned by the recursive call. A good answer to this question is an implementation of subroutines **Modify** and **Fix**. Note that, as in the covering setting, our generic algorithms may use a different type of inequality (weight function) in each recursive call. Moreover, they may use a different property in each recursive call. However, this may require us to implement several versions of subroutines **Modify** and **Fix**. Also, when using a nontrivial recursion base, we can look at the primal-dual (local ratio) phase of the algorithm as a clean-up phase whose output is an instance of a certain type that we know how to solve by subroutine **Base**.

The minimization frameworks can be applied to a large family of algorithms. They can be used in cases of noncovering problems as demonstrated in section 5.3.2 on *minimum* 2-*satisfiability.* They can be used to analyze algorithms that have a nonstandard recursion base, such as the $(2 - \frac{\log\log n}{2\log n})$-approximation algorithm for *vertex cover* from [7], or the 2.5-approximation algorithm for *feedback vertex set in tournaments* given in section 5.3.1. The frameworks can be used to explain algorithms that do not use r-effectiveness with respect to *minimality*, and use a nonstandard instance modification. They can also be used on problems whose solutions are nonboolean. An algorithm using a nonstandard instance modification that approximates a nonboolean *bandwidth trading* problem is given in section 5.3.3. Another example of an algorithm approximating a nonboolean problem is a primal-dual algorithm by Guha et al. [28] for *capacitated vertex cover.* A local ratio interpretation can be found in [5].

Another important point is that an r-effective weight function with respect to a property $\mathcal{P}$ and an r-effective inequality with respect to $\mathcal{P}$ are one and the same. This can be shown in a way similar to the proof of Lemma 4.5. Thus, the equivalence between the two paradigms that was shown with respect to algorithms for covering problems continues to hold even in a more general setting. Namely, Algorithms **LR-min** and **PDmin** are equivalent. We note that the equivalence extends to algorithms outside the scope of our frameworks. For example, in [12] we show that the fractional local ratio technique can be explained using primal-dual arguments.

5.3. Applications.

5.3.1. Feedback vertex set in tournaments. Cai, Deng, and Zang [16] presented a 2.5-approximation algorithm for *feedback vertex set in tournaments* (see section 4.4.3). The algorithm is divided into two parts: a local ratio phase that disposes of certain *forbidden* subtournaments, and an algorithm that finds an optimal solution in any tournament that does not contain these forbidden subtournaments. The forbidden subtournaments are shown in Figure 5.3 below (where the two arcs not shown in T_1 may take any direction).

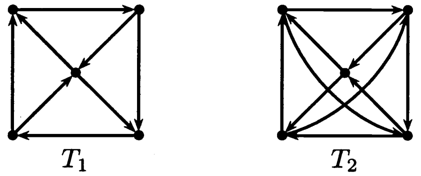


FIG. 5.3.

The local ratio phase employs the following fully 2.5-effective weight function. Let F be a set of five positive-weight vertices inducing a forbidden subtournament and define

$$\delta(v) = \begin{cases} \epsilon, & v \in F, \\ 0 & \text{otherwise,} \end{cases}$$

where $\epsilon = \min_{v \in F} \{w(v)\}$. δ is fully 2.5-effective since the cost of every feasible solution is at most 5ϵ, whereas the minimum weight is at least 2ϵ since every set of four vertices in F contains a triangle. After removing at least one vertex from every forbidden subtournament using local ratio, the problem can be solved optimally on the remaining graph. This algorithm can be seen as an implementation of Algorithm **LR-min** in which subroutines **Modify** and **Fix** are standard, and subroutine **Base** is the algorithm that solves the problem on tournaments that do not contain the forbidden subtournaments.

Using our primal-dual framework, this algorithm can be also analyzed using primal-dual arguments. This can be done by using 2.5-effective inequalities of the form $\sum_{u \in F} x_u \geq 2$, where F is a set of five positive-weight vertices inducing a forbidden subtournament. Clearly, these inequalities are valid with respect to the original instance. Cai, Deng, and Zang [16] show that the integrality gap of the (FVST) (see section 4.4.3) is 1 in the case of tournaments that do not contain the forbidden subtournaments. They actually prove the stronger claim that in tournaments that do not contain the forbidden subtournaments, the primal and dual programs have identical cost integral solutions.

5.3.2. Minimum weight 2-satisfiability. Given a 2CNF formula φ with m clauses on the variables $x_1, \ldots, x_n$ and a weight function w on the variables, the weight of a truth assignment $x \in \{0,1\}^n$ is $\sum_{i=1}^n w_i x_i$. The *minimum weight 2-satisfiability* problem (or min-2SAT for short) is to find a minimum weight truth assignment $x \in \{0,1\}^n$ which satisfies φ, or to determine that no such assignment exists. We formulate min-2SAT as follows:

$$\text{(2SAT)} \qquad \begin{array}{lll} \min & \displaystyle\sum_{i=1}^n w_i x_i & \\ \text{s.t.} & x_i + x_j \geq 1 & \forall (x_i \vee x_j) \in \varphi, \\ & x_i - x_j \geq 0 & \forall (x_i \vee \overline{x}_j) \in \varphi, \\ & -x_i - x_j \geq -1 & \forall (\overline{x}_i \vee \overline{x}_j) \in \varphi, \\ & x_i \in \{0,1\} & \forall i \in \{1, \ldots, n\}. \end{array}$$

Gusfield and Pitt [29] presented an $O(mn)$ time 2-approximation algorithm for min-2SAT. Though they did not use local ratio arguments explicitly, their algorithm can be easily analyzed using local. Hochbaum et al. [32] presented a 2-approximation algorithm for the *two variables per constraint integer programming* problem (2VIP) that generalizes min-2SAT. Later, Bar-Yehuda and Rawitz [9] presented a local ratio 2-approximation algorithm for 2VIP that is more efficient than the algorithm from [32]. On the special case of min-2SAT this algorithm is a variant of the Gusfield–Pitt algorithm. Note that min-2SAT can be approximated using a reduction to vertex cover [30, pp. 131–132].

First, we can check whether φ is satisfiable by using the algorithm from [21]. Thus, we may assume that φ is satisfiable. In order to design a 2-approximation algorithm we need to construct 2-effective inequalities. Given a literal ℓ, let $T(\ell)$ denote the set of variables which must be assigned TRUE whenever ℓ is assigned TRUE. (Constructing $T(\ell)$ for some literal ℓ can be done efficiently by using constraint propagation.) Let x_i, x_j, and x_k be variables such that $x_j \in T(x_i)$ and $x_k \in T(\overline{x}_i)$. For such variables the inequality $x_j + x_k \geq 1$ is valid. Note that one can get inequalities of this form by summing up the appropriate inequalities from the program 2SAT. Moreover, it is not hard to see that this inequality is fully 2-effective. However, instead of using these inequalities one at a time, we can use an inequality of the form

$\sum_{x_j \in T(x_i)} a_j x_j + \sum_{x_k \in T(\overline{x}_i)} b_k x_k \geq \beta$ where all the a_j's and b_k's are nonnegative and $\beta = \sum_j a_j = \sum_k b_k$. This inequality is 2-effective since it is a linear combination of inequalities of the form $x_j + x_k \geq 1$.

Let $\alpha x \geq \beta$ be such an inequality in which $\beta = \min\{\sum_{x_j \in T(x_i)} w_j, \sum_{x_k \in T(\overline{x}_i)} w_k\}$. Assume without loss of generality that $\sum_{x_i \in T(x_1)} w_i \leq \sum_{x_j \in T(\overline{x}_1)} w_j$. Observe that if we subtract α from the objective function, assigning TRUE to all literals in $T(x_i)$ is free of charge. It can be shown that this partial assignment does not change the satisfiability of the formula. That is, if φ' is the formula we get by performing this zero-weight partial assignment to the variables of a formula φ, φ' is satisfiable if and only if φ is satisfiable. After performing this instance modification the rest of the assignment can be found recursively. The primal-dual implementation of the algorithm is as follows. At the recursion base we return an empty assignment on the empty formula. If the formula φ is not empty, we pick a variable x_i and construct an inequality $\alpha x \geq \beta$ as shown above. Note that such inequalities are valid with respect to the original instance. We call subroutine **Modify** that in this case constructs a zero-weight partial assignment for φ and creates a new formula φ'. Then we recursively solve the problem on φ'. Afterwards, subroutine **Fix** combines the assignment for φ' that was returned and the partial assignment that was constructed by subroutine **Modify**. For the local ratio implementation, it is enough to notice that α is a fully 2-effective weight function. (For more details see [9].)

5.3.3. A bandwidth trading problem. Bhatia et al. [15] studied the following *bandwidth trading* problem. We are given a set of machine *types* $\mathcal{T} = \{T_1, \ldots, T_m\}$ and a set of *jobs* $J = \{1, \ldots, n\}$. Each machine type T_i is defined by two parameters: a time interval $I(T_i)$ during which it is *available*, and a weight $w(T_i)$, which represents the weight of allocating a machine of this type. Each job j is defined by a single time interval $I(j)$ during which it must be processed. We say that job j *contains* time t if $t \in I(j)$. A given job j may be *scheduled feasibly* on a machine of type T if type T is available throughout the job's interval, i.e., if $I(j) \subseteq I(T)$. A *schedule* is a set of machines together with an assignment of each job to one of them. It is *feasible* if every job is assigned feasibly and no two jobs with intersecting intervals are assigned to the same machine. The weight of a feasible schedule is the total cost of the machines it uses, where the weight of a machine is defined as the weight associated with its type. The goal is to find a minimum-weight feasible schedule. We assume that a feasible schedule exists. (This can be checked easily.) Bhatia et al. [15] presented a primal-dual 3-approximation algorithm for this problem. A detailed local ratio analysis of their algorithm can be found in [5]. This algorithm constructs weight functions or inequalities that are r-effective weight functions with respect to a property $\mathcal{P}$ different from *minimality* and modifies the solution returned by a recursive call in a rather elaborate manner.

We present the algorithm in local ratio terms in Figure 5.4.

To complete the description of the algorithm we need to describe the transformation of S' to S referred to in line 9. Instead, we just point out two facts relating to this transformation. (The details of the transformation appear in [15] and also in [5].)

1. For all machine types T, S does not use more machines of type T than S'.
2. Let k be the number of jobs containing time t (Line 2). The number of machines used by S whose types are in $\mathcal{T}_t$ is at most $3k$.

Based on these facts, we show that Algorithm **BT** is a specific implementation of Algorithm **LRmin** that returns 3-approximate solutions. By fact 1, $w'(S) \leq w'(S')$, where $w' = w - \delta$, and therefore S is 3-approximate with respect to w'. Thus,

```
Algorithm BT($\mathcal{T}, J, w$).
1.  If $J = \emptyset$, return $\emptyset$
2.  Let $t$ be a point in time contained in a maximum number of jobs,
    and let $\mathcal{T}_t$ be the set of machine types available at time $t$
3.  Let $\epsilon = \min\{w(T) : T \in \mathcal{T}_t\}$
4.  Define the weight function $\delta(T) = \begin{cases} \epsilon, & T \in \mathcal{T}_t, \\ 0 & \text{otherwise}, \end{cases}$

    /* Subroutine Modify */
5.  Let $\mathcal{T}'_t = \{T : T \in \mathcal{T}_t,\ w(T) = \delta(T)\}$
6.  Let $J' = \{j \in J : \exists T \in \mathcal{T}'_t,\ I(j) \subseteq I(T)\}$
7.  $S' \leftarrow$ BT($\mathcal{T} \setminus \mathcal{T}'_t, J \setminus J', w - \delta$)
    /* Subroutine Fix */
8.  Extend $S'$ to $J$ by allocating $|J'|$ machines and scheduling one
    job from $J'$ on each.
    Job $j \in J'$ is assigned to a machine of type $T \in \mathcal{T}'_t$
    such that $I(j) \subseteq I(T)$.
9.  Transform $S'$ into a new schedule $S$ as described below
10. Return $S$
```

Fig. 5.4.

subroutines **Modify** and **Fix** work as required. (Subroutine **Base** is standard in this case.) By fact 2, $\delta(S) \leq 3k\epsilon$, and because there are k jobs containing time t—each of which can be scheduled only on machines whose types are in $\mathcal{T}_t$, and no two of which may be scheduled on the same machine—the optimum cost is at least $k\epsilon$. Thus, S is 3-approximate with respect to δ.

Bhatia et al. [15] formulated the bandwidth trading problem by the following program:

$$\begin{array}{lll} \min & \sum_{i=1}^{n} w(T_i)x_i & \\ \text{s.t.} & \sum_{i} y_{ij} \geq 1 & \forall j \in J, \\ & x_i - \sum_{j \in J(t)} y_{ij} \geq 0 & \forall T_i \in \mathcal{T},\ \forall t \in E \cap I(T_i), \\ & x_i \in \mathbb{N} & \forall T_i \in \mathcal{T}, \\ & y_{ij} \in \{0,1\} & \forall T_i \in \mathcal{T}, j \in J, \end{array}$$

where

- x_i represents the number of machines allocated of type T_i;
- $y_{ij} = 1$ if and only if job j is assigned to machine type T_i; note that y_{ij} is defined only if $I(j) \subseteq I(T_i)$, where i is of type T;
- E is the set of endpoints of job intervals;
- $J(t) = \{j : t \in I(j)\}$.

In order to transform Algorithm **BT** into a primal-dual algorithm, we use the inequality $\delta \cdot x \geq k\epsilon$. It is not hard to verify that this version of Algorithm **BT** is an implementation of Algorithm **PDmin**. The above inequality is valid with respect to the original instance, since if there are k jobs whose interval contains time t, then at least k machines whose types belong to T_t must be allocated.

We remark that our primal-dual analysis is slightly different from the analysis in [15]. Specifically, their algorithm uses similar but not identical inequalities that can be described as linear combinations of inequalities from the above formulation.

6. Maximization problems.

Bar-Noy et al. [3] developed constant factor approximation algorithms for various resource allocation and scheduling problems using local ratio. They also presented primal-dual algorithms for these problems. This was the first time a local ratio or primal-dual approximation algorithm for a natural maximization problem was presented. In this section we present two equivalent generic approximation algorithms for maximization problems that can be used to analyze the algorithms from [3]. We demonstrate this on one of the problems that was discussed in [3] called *interval scheduling.* Also, we show that our generic algorithms can explain the exact optimization (or 1-approximation) algorithm for the *longest path in a DAG* problem.

6.1. The frameworks.

Before describing the generic algorithms, we address the issue of r-effectiveness in the context of maximization. We discuss the issue in terms of weight functions, but a similar discussion can be made in terms of inequalities. Recall that δ is r-effective with respect to a property $\mathcal{P}$ if there exists β such that $\beta \leq \delta x \leq r\beta$ for every solution x that satisfies $\mathcal{P}$. In the maximization setting it is more convenient to consider the following equivalent definition. δ is r-effective with respect to a property $\mathcal{P}$ if there exists β such that $\frac{\beta}{r} \leq \delta x \leq \beta$ for every solution x that satisfies $\mathcal{P}$. Clearly any feasible solution that satisfies $\mathcal{P}$ is r-approximate with respect to δ.

Our frameworks are recursive and work as follows. If the instance is empty, then return the empty set. Otherwise, construct a weight function (inequality) that is r-effective with respect to some property $\mathcal{P}$. Subtract the weight function (coefficients of inequality) from the objective function. Remove some of the nonpositive-weight elements from the instance. (The decision of which element to remove depends on the problem at hand. Algorithms for *packing* problems usually remove all nonpositive-weight elements.) Then recursively solve the problem with respect to the new instance and weights. Upon returning from the recursive call, the solution returned is fixed such that it satisfies $\mathcal{P}$. We remark that in order to simplify the presentation our maximization algorithms are not as general as our minimization algorithms. Namely, they use a limited version of subroutine **Modify** that simply removes some nonpositive-weight elements from the instance and do not use a version of subroutine **Base** at all. We also limit our discussion in this section to sets of feasibility constraints $\mathcal{F}$ for which $x \in \{0,1\}^n$.

We start with our local ratio approximation algorithm for maximization problems —Algorithm **LRmax** (see Figure 6.1). The initial call is **LRmax**$(\{1,\ldots,n\},w)$. A recursive call of Algorithm **LRmax** considers the instance that is induced by the set of elements N that corresponds to the set of positive-weight elements. It starts with the construction of a weight function δ. Then a recursive call is made on the instance that is induced by the objective function $w-\delta$ and the set $N \setminus N^-$, where N^- is a set that contains nonpositive-weight elements with respect to $w-\delta$. Subroutine **Fix** is used to fix the solution returned by adding only zero-weight elements with respect to $w-\delta$. The resulting solution satisfies property $\mathcal{P}$.

```
Algorithm LRmax$(N, w)$.
    If $N = \emptyset$, return $\emptyset$
    Construct a weight function $\delta$ which is $r$-effective
            with respect to $(\mathcal{F}, N)$ and $\mathcal{P}$
    Let $N^- \subseteq \{j : w_j - \delta_j \leq 0\}$
    $x' \leftarrow$ LRmax$(N \setminus N^-, w - \delta)$
    $x \leftarrow$ Fix$(\mathcal{F}, w - \delta, x, \mathcal{P})$
    Return $x$
```

Fig. 6.1.

```
Algorithm PDmax$(N, w)$.
    If $N = \emptyset$, return $\emptyset$
    Construct a valid inequality $\alpha^k x \leq \beta^k$ which is $r$-effective
            with respect to $(\mathcal{F}, N)$ and $\mathcal{P}$
    Let $N^- \subseteq \{j : w_j - \alpha_j \leq 0\}$
    $x' \leftarrow$ PDmax$(N \setminus N^-, w - \alpha)$
    $x \leftarrow$ Fix$(\mathcal{F}, w - \alpha, x, \mathcal{P})$
    Return $x$
```

Fig. 6.2.

We prove by induction that Algorithm **LRmax** returns an r-approximate solutions with respect to (N, w). In the base case, $\emptyset$ is an optimal solution. For the inductive step, examine x at the end of the recursive call. By the induction hypothesis x' is r-approximate with respect to $(N \setminus N^-, w - \delta)$. Moreover, since $w_j - \delta_j \leq 0$ for every $j \in N^-$, x is r-approximate with respect to $(N, w - \delta)$. (Recall that subroutine **Fix** adds only zero-weight elements with respect to $w - \delta$.) Finally, x satisfies $\mathcal{P}$ due to subroutine **Fix**; therefore by the r-effectiveness of δ with respect to $\mathcal{P}$ and by the local ratio theorem, we get that x is r-approximate with respect to (N, w) as well.

Algorithm **PDmax** (see Figure 6.2) is very similar to Algorithm **LRmax**. Obviously, Algorithm **PDmax** uses inequalities instead of weight functions. Also, as in the minimization case, we assume that the inequalities that are used by the algorithm are valid with respect to the original set of constraints $\mathcal{F}$. This condition is imperative to the construction of a feasible dual solution.

We show that Algorithm **PDmax** returns r-approximate solutions. Let notation with subscript k denote the appropriate object in the kth iteration, and let $t + 1$ be the recursion depth. Consider the linear program

$$\text{(P)} \qquad \begin{array}{lll} \min & wx & \\ \text{s.t.} & \alpha^k x \leq \beta^k & k \in \{1, \ldots, t\}, \\ & x \geq 0, & \end{array}$$

where $\alpha^k x \leq \beta^k$ is the inequality used in the kth recursive call. Every feasible solution satisfies the constraints in (P), namely, $\textsc{Sol}(\mathcal{F}) \subseteq \textsc{Sol}(\text{P})$. Thus, $x \in \textsc{Sol}(\text{P})$ and $\textsc{Opt}(\text{P}) \geq \textsc{Opt}(\mathcal{F}, w)$.

Consider the dual of (P):

$$\text{(D)} \qquad \begin{array}{ll} \min & \displaystyle\sum_{k=1}^{t} \beta^k y_k \\ \text{s.t.} & \displaystyle\sum_{k=1}^{t} \alpha_j^k y_k \geq w_j \quad j \in \{1, \dots, n\}\,, \\ & y \geq 0. \end{array}$$

We claim that $y = (1, \dots, 1)$ is a feasible solution to (D). To do that we conceptually add the following between line 2 and line 3: $y_k \leftarrow 1$. Clearly, the resulting dual solution is $y = (1, \dots, 1)$. In terms of the dual solution, elements leave the set N only when their corresponding dual constraint is satisfied. Algorithm **PDmax** terminates when the current instance is empty, namely, when $N = \emptyset$. Therefore, at termination all dual constraints are satisfied.

We prove by induction that $w^k x^k \geq \frac{1}{r} \sum_{l \geq k} y_l \beta^l$. At the induction basis, $0 = w^{t+1} x^{t+1} \geq \frac{1}{r} \sum_{l \geq t+1} y_l \beta^l = 0$. For $k \leq t$ we have

$$w^k x^k = (w^{k+1} + \alpha^k) x^k = w^{k+1} x^{k+1} + y_k \alpha^k x^k \geq \frac{1}{r} \cdot \sum_{l \geq k+1} y_l \beta^l + \frac{\beta^k}{r} = \frac{1}{r} \cdot \sum_{l \geq k} y_l \beta^l,$$

where the second equality is due to the fact that subroutine **Fix** uses only zero-weight elements, and the inequality is implied by the induction hypothesis and the r-effectiveness of the kth inequality. Therefore, $wx = w^1 x^1 \geq \frac{1}{r} \sum_{l \geq 1} y_l \beta^l \geq \frac{1}{r} \cdot \textsc{Opt}(\text{P}) \geq \frac{1}{r} \cdot \textsc{Opt}(\mathcal{F}, w)$.

Notice that the maximization case is different from the minimization case. In the latter we keep the weights nonnegative, while in the former, weights are allowed to be negative. Moreover, the objective function in the maximization case is expected to be nonpositive when the algorithm terminates. This means, in primal-dual terms, that the dual solution is initially not feasible, and its feasibility is improved during the execution of the algorithm. Also, at termination, the negative entries of the weight function correspond to the nontight dual constraints. This difference makes life more complicated in the maximization setting. Speaking in local ratio terms, in the minimization case, the weight function δ is constructed such that it satisfies two conditions: (1) $\delta \leq w$, and (2) there exists an element j for which $w_j = \delta_j$. In the maximization case, the second condition is satisfied but the first is not. In fact, given an r-effective weight function δ, it is not always clear by which factor $\epsilon > 0$ we should multiply it before subtracting it from the objective function. We are allowed to increase ϵ as long as the solution returned by the recursive call can be fixed using only zero-weight elements.

6.2. Applications.

6.2.1. Interval scheduling. As mentioned before, in [3] Bar-Noy et al. presented local ratio approximation algorithms for several resource allocation and scheduling problems that can be explained by our frameworks. We demonstrate this by analyzing one of the algorithms from [3] that approximates a problem called *interval scheduling*. Bar-Noy et al. also presented primal-dual algorithms for the same problems. However, in order to do so they modified the original algorithms. We show that there is no need to change the algorithms in order to supply a primal-dual analysis.

In the interval scheduling problem we are given a set of *activities*, each requiring the utilization of a given *resource*. The activities are specified as a collection of sets $\mathcal{A}_1, \ldots, \mathcal{A}_m$. Each set represents a single activity: it consists of all possible *instances* of that activity. An instance $I \in \mathcal{A}_i$ is defined by the following parameters:

1. A half-open time interval $[s(I), e(I))$ during which the activity will be executed. $s(I)$ and $e(I)$ are called the *start-time* and *end-time* of the instance.
2. The weight $w(I) \geq 0$ gained by scheduling this instance of the activity.

A *schedule* is a collection of instances. It is feasible if it contains at most one instance of every activity and at most one instance for all time instants t. In the interval scheduling problem our goal is to find a schedule that maximizes the total weight accrued by instances in the schedule.

The interval scheduling problem can be formulated by means of an integer program on the boolean variables $\{x_I : I \in \mathcal{A}_i, 1 \leq i \leq m\}$.

$$\begin{array}{lll} \max & \sum_I w(I) x_I & \\ \text{s.t.} & \sum_{I: s(I) \leq t < e(I)} x_I \leq 1 & \forall t, \\ & \sum_{I: I \in \mathcal{A}_i} x_I \leq 1 & \forall i \in \{1, \ldots, m\}, \\ & x_I \in \{0, 1\} & \forall i \; \forall I \in A_i. \end{array}$$

The 2-approximation algorithm for interval scheduling from [3] can be viewed as an application of Algorithm **LRmax**. In order to describe it as such, we need to show (1) how to construct a weight function δ that is 2-effective with respect to some property $\mathcal{P}$; (2) which elements are removed from the instance (i.e., which elements are taken into N^-); and (3) how to fix the solution returned by the recursive call (i.e., describe subroutine **Fix**). Let J be an instance with minimum end-time, and let $\mathcal{A}(J)$ and $\mathcal{I}(J)$ be the activity to which instance J belongs and the set of instances intersecting J (including J), respectively. (See Figure 6.3.) Define

$$\delta(I) = \begin{cases} w(J), & I \in \mathcal{A}(J) \cup \mathcal{I}(J), \\ 0 & \text{otherwise.} \end{cases}$$

We show that δ is 2-effective with respect to some property $\mathcal{P}$. We say that a feasible schedule S is J-maximal if either it contains J or J cannot be added to S without rendering it infeasible. It is not hard to verify that the weight of every J-maximal schedule with respect to δ is at least $w(J)$ and no more than $2 \cdot w(J)$. (Notice that a feasible schedule contains no more than two instances from $\mathcal{A}(J) \cup \mathcal{I}(J)$.) Now, the elements that are taken into N^- are all nonpositive elements with respect to $w - \delta$. Finally, we describe subroutine **Fix**. Let S' be the schedule returned by the recursive call. If $S' \cup \{J\}$ is a feasible solution, return $S = S' \cup \{J\}$. Otherwise, return $S = S'$. Clearly, S is J-maximal.

As mentioned before, Bar-Noy et al. [3] also presented primal-dual algorithms that are slightly different from their local ratio algorithms. In terms of the interval scheduling problem they modified the original algorithm by using a different 2-effective weight function:

$$\delta'(I) = \begin{cases} w(J), & I = J, \\ \frac{1}{2} w(J), & I \in \mathcal{A}(J) \cup \mathcal{I}(J) \setminus \{J\}, \\ 0 & \text{otherwise.} \end{cases}$$

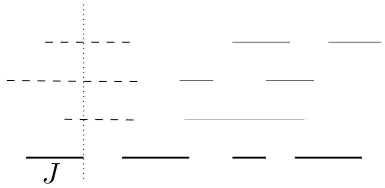


FIG. 6.3. *$J, \mathcal{A}(J), \mathcal{I}(J)$: heavy lines represent $A(J)$, dashed lines represent $\mathcal{I}(J)$.*

The corresponding inequality is $\frac{1}{2}\sum_{I\in\mathcal{I}(J)} x_I + \frac{1}{2}\sum_{I\in A(J)} x_I \leq 2$. Note that this inequality is a linear combination of two inequalities from the above integer program. The original algorithm can be explained by the 2-effective inequality $\sum_{I\in\mathcal{I}(J)\cup\mathcal{A}(J)} x_I \leq 2$. The difference between δ and δ' (or between their corresponding inequalities) is the ratio between the weight of J and the weights of the other instances in $\mathcal{A}(J)\cup\mathcal{I}(J)$. In fact, any value between 1 and 2 is acceptable.

6.2.2. Longest path in a DAG. The *longest path* problem is as follows: given an arc-weighted directed graph $G = (V, A)$ and two distinguished vertices s and t, find a simple path from s to t of maximum *length*, where the *length* of a path is defined as the sum of weights of its arcs. For general graphs (either directed or undirected) the problem is NP-hard [24], but for *directed acyclic graphs* (DAGs) it is solvable in linear time by a Dijkstra-like algorithm that processes the nodes in topological order. The problem of finding the longest path in a DAG (also called the *critical path*) arises in the context of PERT (Program Evaluation and Review Technique) charts. For more details see [18, p. 538] or [20, pp. 138–142].

We show that the above-mentioned linear time algorithm can be seen as an implementation of Algorithms **LRmax** and **PDmax**. We allow negative arc weights, and we assume that every vertex is reachable from s. (Otherwise, simply delete all vertices that are unreachable from s.) We also assume that the vertices of G were topologically sorted, and that t is the last vertex in this topological sort. Instead of solving the original problem we solve the following more general problem. Namely, instead of searching for a longest path from s to t we would like to find the longest path from some vertex in a set S to t without using arcs within S. In the original problem $S = \{s\}$. Also, if $s \in S$ and for all $u \in S$ the longest path from s to u is of length zero, then the problem is equivalent to the original problem.

Consider a cut $(S, \bar{S})$ such that $s \in S$, $t \in \bar{S}$, and there is no arc leaving $\bar{S}$ and entering S. Note that if we take the first k vertices in the topological sort, we get such a cut. We define the following function:

$$\delta(e) = \begin{cases} \epsilon, & e \in S \times \bar{S}, \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, any path from s to t must cross the cut $(S, \bar{S})$ exactly once, and thus δ is fully 1-effective. Equivalently, the equality $\sum_{e\in S\times\bar{S}} x_e = 1$ is valid. Having defined a suitable weight function or equality, we continue with a description of the algorithm. We describe a recursive call of the algorithm using local ratio terms. Let v be the vertex which is the first in $\bar{S}$ according to the topological sort. Let $\epsilon = \max_{u\in S}\{w(u, v)\}$ (ϵ may be negative), and let $e = (u, v)$ be an arc such that $u \in S$ and $w(u, v) = \epsilon$. If $v = t$, then return a path containing u and t. Otherwise, solve the problem recursively on $(G, S \cup \{v\}, w - \epsilon \cdot \delta)$. Now, let $v_1, \ldots, v_\ell$ be the path returned. If $v_1 = v$, then return the path $u, v_1, \dots, v_\ell$; otherwise return $v_1, \dots, v_\ell$.

Acknowledgments. We thank Guy Even, Eran Mann, Seffi Naor, and especially Ari Freund for helpful comments and discussions.

REFERENCES

[1] A. Agrawal, P. Klein, and R. Ravi, *When trees collide: An approximation algorithm for the generalized Steiner problem on networks*, SIAM J. Comput., 24 (1995), pp. 440–456.

[2] V. Bafna, P. Berman, and T. Fujito, *A 2-approximation algorithm for the undirected feedback vertex set problem*, SIAM J. Discrete Math., 12 (1999), pp. 289–297.

[3] A. Bar-Noy, R. Bar-Yehuda, A. Freund, J. Naor, and B. Shieber, *A unified approach to approximating resource allocation and scheduling*, J. ACM, 48 (2001), pp. 1069–1090.

[4] R. Bar-Yehuda, *One for the price of two: A unified approach for approximating covering problems*, Algorithmica, 27 (2000), pp. 131–144.

[5] R. Bar-Yehuda, K. Bendel, A. Freund, and D. Rawitz, *Local ratio: A unified framework for approximation algorithms*, ACM Comput. Surveys, 36 (2004), pp. 422–463.

[6] R. Bar-Yehuda and S. Even, *A linear time approximation algorithm for the weighted vertex cover problem*, J. Algorithms, 2 (1981), pp. 198–203.

[7] R. Bar-Yehuda and S. Even, *A local-ratio theorem for approximating the weighted vertex cover problem*, Ann. Discrete Math., 25 (1985), pp. 27–46.

[8] R. Bar-Yehuda, M. M. Halldórsson, J. Naor, H. Shachnai, and I. Shapira, *Scheduling split intervals*, in Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2002, pp. 732–741.

[9] R. Bar-Yehuda and D. Rawitz, *Efficient algorithms for bounded integer programs with two variables per constraint*, Algorithmica, 29 (2001), pp. 595–609.

[10] R. Bar-Yehuda and D. Rawitz, *On the equivalence between the primal-dual schema and the local ratio technique*, in Proceedings of the 4th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, Lecture Notes in Comput. Sci. 2129, Springer-Verlag, Berlin, 2001, pp. 24–35.

[11] R. Bar-Yehuda and D. Rawitz, *Local ratio with negative weights*, Oper. Res. Lett., 32 (2004), pp. 540–546.

[12] R. Bar-Yehuda and D. Rawitz, *Using fractional primal-dual to schedule split intervals with demands*, in Proceedings of the 13th Annual European Symposium on Algorithms, Lecture Notes in Comput. Sci. 3669, Springer-Verlag, Berlin, 2005, pp. 714–725.

[13] A. Becker and D. Geiger, *Optimization of Pearl's method of conditioning and greedy-like approximation algorithms for the vertex feedback set problem*, Artificial Intelligence, 83 (1996), pp. 167–188.

[14] D. Bertsimas and C.-P. Teo, *From valid inequalities to heuristics: A unified view of primal-dual approximation algorithms in covering problems*, Oper. Res., 46 (1998), pp. 503–514.

[15] R. Bhatia, J. Chuzhoy, A. Freund, and J. Naor, *Algorithmic aspects of bandwidth trading*, in Proceedings of the 30th International Colloquium on Automata, Languages, and Programming, Lecture Notes in Comput. Sci. 2719, Springer-Verlag, Berlin, 2003, pp. 751–766.

[16] M.-C. Cai, X. Deng, and W. Zang, *An approximation algorithm for feedback vertex sets in tournaments*, SIAM J. Comput., 30 (2001), pp. 1993–2007.

[17] F. A. Chudak, M. X. Goemans, D. S. Hochbaum, and D. P. Williamson, *A primal-dual interpretation of recent 2-approximation algorithms for the feedback vertex set problem in undirected graphs*, Oper. Res. Lett., 22 (1998), pp. 111–118.

[18] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1990.

[19] G. B. Dantzig, L. R. Ford, and D. R. Fulkerson, *A primal-dual algorithm for linear programs*, in Linear Inequalities and Related Systems, H. W. Kuhn and A. W. Tucker, eds., Princeton University Press, Princeton, NJ, 1956, pp. 171–181.

[20] S. Even, *Graph Algorithms*, Computer Science Press, Woodland Hills, CA, 1979.

[21] S. Even, A. Itai, and A. Shamir, *On the complexity of timetable and multicommodity flow problems*, SIAM J. Comput., 5 (1976), pp. 691–703.

[22] A. Freund and D. Rawitz, *Combinatorial interpretations of dual fitting and primal fitting*, in Proceedings of the 1st Workshop on Approximation and Online Algorithms, Lecture Notes in Comput. Sci. 2909, Springer-Verlag, Berlin, 2003, pp. 137–150.

[23] T. Fujito, *A unified approximation algorithm for node-deletion problems*, Discrete Appl. Math., 86 (1998), pp. 213–231.

[24] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, CA, 1979.

[25] M. X. Goemans, A. V. Goldberg, S. Plotkin, D. B. Shmoys, É. Tardos, and D. P. Williamson, *Improved approximation algorithms for network design problems*, in Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 1994, pp. 223–232.

[26] M. X. Goemans and D. P. Williamson, *A general approximation technique for constrained forest problems*, SIAM J. Comput., 24 (1995), pp. 296–317.

[27] M. X. Goemans and D. P. Williamson, *The primal-dual method for approximation algorithms and its application to network design problems*, in Approximation Algorithms for NP-Hard Problems, D. S. Hochbaum, ed., PWS Publishing, Boston, MA, 1997, Chap. 4., pp. 144–191.

[28] S. Guha, R. Hassin, S. Khuller, and E. Or, *Capacitated vertex covering with applications*, in Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2002, pp. 858–865.

[29] D. Gusfield and L. Pitt, *A bounded approximation for the minimum cost* 2*-SAT problem*, Algorithmica, 8 (1992), pp. 103–117.

[30] D. S. Hochbaum, ed., *Approximation Algorithms for NP-Hard Problems*, PWS Publishing, Boston, MA, 1997.

[31] D. S. Hochbaum, *Solving integer programs over monotone inequalities in three variables: A framework of half integrality and good approximations*, European J. Oper. Res., 140 (2002), pp. 291–321.

[32] D. S. Hochbaum, N. Megiddo, J. Naor, and A. Tamir, *Tight bounds and* 2*-approximation algorithms for integer programs with two variables per inequality*, Math. Program., 62 (1993), pp. 69–83.

[33] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. Vazirani, *Greedy facility location algorithms analyzed using dual-fitting with factor-revealing LP*, J. ACM, 50 (2003), pp. 795–824.

[34] K. Jain and V. V. Vazirani, *Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation*, J. ACM, 48 (2001), pp. 274–296.

[35] H. Karloff, *Linear Programming*, Progr. Theoret. Comput. Sci., Birkhäuser Boston, Boston, MA, 1991.

[36] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, 5th ed., Prentice-Hall, Englewood Cliffs, NJ, 1982.

[37] R. Ravi and P. Klein, *When cycles collapse: A general approximation technique for constrained two-connectivity problems*, in Proceedings of the 3rd MPS Conference on Integer Programming and Combinatorial Optimization, CIACO, Louvain-la-Neuve, Belgium, 1993, pp. 39–56.

[38] V. V. Vazirani, *Approximation Algorithms*, 2nd ed., Springer-Verlag, Berlin, 2001.

[39] D. P. Williamson, *The primal dual method for approximation algorithms*, Math. Program., 91 (2002), pp. 447–478.

[40] D. P. Williamson, M. X. Goemans, M. Mihail, and V. V. Vazirani, *A primal-dual approximation algorithm for generalized Steiner network problems*, Combinatorica, 15 (1995), pp. 435–454.

SIAM J. DISCRETE MATH.
Vol. 19, No. 3, pp. 798–806

INTEGER DECOMPOSITION FOR POLYHEDRA DEFINED BY NEARLY TOTALLY UNIMODULAR MATRICES[*]

DION GIJSWIJT[†]

Abstract. We call a matrix A *nearly totally unimodular* if it can be obtained from a totally unimodular matrix $\tilde{A}$ by adding to each row of $\tilde{A}$ an integer multiple of some fixed row a^T of $\tilde{A}$. For an integer vector b and a nearly totally unimodular matrix A, we denote by $P_{A,b}$ the integer hull of the set $\{x \in \mathbb{R}^n \mid Ax \leq b\}$. We show that $P_{A,b}$ has the integer decomposition property and that we can find a decomposition of a given integer vector $x \in kP_{A,b}$ in polynomial time.

An interesting special case that plays a role in many cyclic scheduling problems is when A is a circular-ones matrix. In this case, we show that given a nonnegative integer k and an integer vector x, testing if $x \in kP_{A,b}$ and finding a decomposition of x into k integer vectors in $P_{A,b}$ can be done in time $O(n(n+m) + \text{size}(x))$, where A is an $m \times n$ matrix. We show that the method unifies some known results on coloring circular arc graphs and edge coloring nearly bipartite graphs. It also gives an efficient algorithm for a packet scheduling problem for smart antennas posed by Amaldi, Capone, and Malucelli in [*Fourth ALIO/EURO Workshop on Applied Combinatorial Optimization*, Pucón, Chile, 2002]; [*Proceedings of the Second Cologne-Twente Workshop on Graphs and Combinatorial Optimization*, Vol. 1, 2003, pp. 1–4].

Key words. integer decomposition, totally unimodular, circular arc graph, nearly bipartite graph, cyclic scheduling, coloring

AMS subject classifications. 90C10, 05C15, 05C70

DOI. 10.1137/S089548010343569X

1. Introduction.

The main motivation for this paper is a very nice combinatorial problem from the field of telecommunications. This problem was posed by Amaldi, Capone, and Malucelli in [1, 2] and concerns scheduling packets for smart antennas. In brief, the problem comes down to the following. Given a finite set V of points on the unit circle and an angle α, find a coloring of V using a minimal number of colors. The coloring restriction is that no segment of length α may contain more than two points of the same color.

We show that this problem as well as some other coloring problems in the literature (see [7, 9, 10, 12]) can be formulated as an integer decomposition problem for polyhedra defined by matrices that are very close to being totally unimodular. This motivates the following definitions. In what follows, all vectors will be column-vectors. We call a matrix A *nearly totally unimodular* if there exists a totally unimodular matrix $\binom{\tilde{A}}{a^\mathsf{T}}$ and an integer vector c such that $A = \tilde{A} + ca^\mathsf{T}$. If A is an $m \times n$ nearly totally unimodular matrix and $b \in \mathbb{Z}^m$ is a vector, we define the integer polyhedron $P_{A,b}$ by

$$P_{A,b} := \text{conv.hull}(\{x \in \mathbb{Z}^n \mid Ax \leq b\}). \tag{1}$$

In section 2 we show that the polyhedron $P_{A,b}$ has the integer decomposition property. That is, every integer vector in $kP_{A,b}$ is the sum of k integer vectors in $P_{A,b}$. The proof reduces the problem of decomposing an integer vector $x \in kP_{A,b}$ to a number of integer

[*] Received by the editors October 1, 2003; accepted for publication (in revised form) May 9, 2005; published electronically December 7, 2005.

http://www.siam.org/journals/sidma/19-3/43569.html

[†] Department of Mathematics, University of Amsterdam, Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands (gijswijt@science.uva.nl).

linear programs with totally unimodular constraint matrix $\binom{\tilde{A}}{a^\mathsf{T}}$. These can be solved in polynomial time by using the ellipsoid method (see [8]). However, in particular instances there may be a more efficient combinatorial algorithm to solve these linear programs. In section 3 we consider the case where A is a circular-ones matrix. We show that in this case a decomposition of x can be found in time $O(n(n+m)+\text{size}(x))$ when A is an $m \times n$ matrix. In section 4 we treat the packet scheduling problem in more detail and apply the results from sections 2 and 3 to obtain an efficient algorithm that solves the packet scheduling problem. We also give some applications to edge coloring nearly bipartite graphs and to coloring proper circular arc graphs.

2. Integer decomposition.

Let A be an $m \times n$ nearly totally unimodular matrix. We assume that A is given as $A = \tilde{A} + ca^\mathsf{T}$ for an integer vector $c \in \mathbb{Z}^m$ and a totally unimodular matrix $\binom{\tilde{A}}{a^\mathsf{T}}$. Let $b \in \mathbb{Z}^m$ be an integer vector. A basic observation is the following.

PROPOSITION 1. *For any integer s, the polyhedron $P_{A,b} \cap \{x \in \mathbb{R}^n \mid a^\mathsf{T}x = s\}$ has the integer decomposition property.*

Proof. First observe that since the matrix $\binom{\tilde{A}}{a^\mathsf{T}}$ is totally unimodular, the polyhedron

$$P := \{x \in \mathbb{R}^n \mid \tilde{A}x \le b - sc,\ a^\mathsf{T}x = s\} \tag{2}$$

is integer. Furthermore, by the well-known theorem of Baum and Trotter characterizing totally unimodularity (see [3]), P has the integer decomposition property. Since P is integer, we have

$$P \subseteq \text{int.hull}(P) \subseteq P_{A,b} \cap \{x \in \mathbb{R}^n \mid a^\mathsf{T}x = s\} \subseteq P, \tag{3}$$

showing that $P = P_{A,b} \cap \{x \in \mathbb{R}^n \mid a^\mathsf{T}x = s\}$, which concludes the proof. □

We can now prove that also $P_{A,b}$ has the integer decomposition property.

THEOREM 1. *Let k be a nonnegative integer and let $x \in \mathbb{Z}^n$. Write $a^\mathsf{T}x = qk + r$ for integers q and r with $0 \le r \le k-1$. Then the following are equivalent:*

(i) $x \in kP_{A,b}$,

(ii) the system

$$\begin{aligned} Ay &\le rb, \\ A(x-y) &\le (k-r)b, \\ a^\mathsf{T}y &= r(q+1) \end{aligned} \tag{4}$$

is feasible,

(iii) $x = x_1 + x_2 + \cdots + x_k$ for integer vectors $x_1, \ldots, x_k \in P_{A,b}$.

In particular, $P_{A,b}$ has the integer decomposition property.

Proof. It is clear that (iii) implies (i). To show that (i) implies (ii), suppose that $\frac{1}{k}x \in P_{A,b}$. Since the polyhedron $P_{A,b}$ is integer, we can write $\frac{1}{k}x = \frac{r}{k}x' + \frac{k-r}{k}x''$, where $x', x'' \in P_{A,b}$ and $a^\mathsf{T}x' = q+1$, $a^\mathsf{T}x'' = q$. Indeed, for a suitably large integer M we can write

$$M \cdot x = \sum_{i=1}^{kM} x_i, \tag{5}$$

where $x_i \in P_{A,b}$ and $a^{\mathsf{T}} x_i \in \mathbb{Z}$ for $i = 1, \dots, kM$ (since we can take the x_i to be integer). Now take such a representation of $M \cdot x$ that minimizes

$$\sum_{i=1}^{kM} (a^{\mathsf{T}} x_i)^2. \tag{6}$$

Then $|a^{\mathsf{T}} x_i - a^{\mathsf{T}} x_j| \leq 1$ for any i and j, since otherwise we can replace x_i and x_j by $\lambda x_i + (1-\lambda)x_j$ and $\lambda x_j + (1-\lambda)x_i$, where $\lambda = 1/|a^{\mathsf{T}} x_i - a^{\mathsf{T}} x_j|$ thus reducing (6). Hence $a^{\mathsf{T}} x_i \in \{q, q+1\}$ for each $i = 1, \dots, n$, and setting

$$x' := \frac{1}{M} \sum_{i \mid a^{\mathsf{T}} x_i = q+1} x_i \quad \text{and} \quad x'' := \frac{1}{M} \sum_{i \mid a^{\mathsf{T}} x_i = q} x_i \tag{7}$$

gives the required decomposition. It follows that x' satisfies (4).

To show that (ii) implies (iii), suppose that the system (4) is feasible. Observe that (4) is equivalent to

$$\begin{aligned} \tilde{A} y &\leq r(b - (q+1)c), \\ \tilde{A} y &\geq \tilde{A} x + (k-r)(qc - b), \\ a^{\mathsf{T}} y &= r(q+1). \end{aligned} \tag{8}$$

Hence (4) has an integer solution y because the matrix $\binom{\tilde{A}}{a^{\mathsf{T}}}$ is totally unimodular. Since y is an integer vector in $r(P_{A,b} \cap \{x \in \mathbb{R}^n \mid a^{\mathsf{T}} x = q+1\})$, we obtain by Proposition 1 a decomposition $y = y_1 + \cdots + y_r$ of y into r integer vectors in $P_{A,b}$. Similarly, Proposition 1 gives a decomposition $x - y = x_1 + \cdots + x_{k-r}$ of $x - y$ into $k - r$ integer vectors $x_1, \dots, x_{k-r}$ in $P_{A,b}$. Hence $x = y_1 + \cdots + y_k + x_1 + \cdots + x_{k-r}$ is the required decomposition of x. □

From Theorem 1 it follows that testing membership of $P_{A,b}$ can be done in polynomial time, since checking feasibility of (4) can be done in polynomial time. Finding the required decompositions of y and $x - y$ can be done in polynomial time. Indeed, denote

$$P := \{z \mid \tilde{A} z \leq b,\ a^{\mathsf{T}} z = s\}. \tag{9}$$

Decomposing an integer vector $y \in rP$ can be done by solving $r - 1$ linear programs, since a decomposition $y = y_1 + y_2$ into integer vectors $y_1 \in P$ and $y_2 \in (r-1)P$ can be found by solving $\{\tilde{A} y_1 \leq b,\ \tilde{A}(y - y_1) \leq (r-1)b,\ a^{\mathsf{T}} y_1 = s\}$.

With a little more care (as was pointed out by an anonymous referee), a decomposition can be found in polynomial time as follows (see [6, 8]). First, $t \leq n+1$ affinely independent integer vectors $y_1, \dots, y_t \in P$ and nonnegative numbers $\lambda_1, \dots, \lambda_t$ with $\lambda_1 + \cdots + \lambda_t = 1$ can be found such that $\frac{1}{r} y = \lambda_1 y_1 + \cdots + \lambda_t y_t$ (algorithmic version of Carathéodory's theorem). Then $y' := y - \lfloor r\lambda_1 \rfloor y_1 - \cdots - \lfloor r\lambda_t \rfloor y_t$ is an integer vector in $r'P$, where $r' = r - \lfloor r\lambda_1 \rfloor - \cdots - \lfloor r\lambda_t \rfloor < t$. Hence y' can be decomposed into r' integer vectors in P by solving less than t linear programs as above.

However, often A and b come from a combinatorial problem that allows more efficient ways of computing a decomposition of x. In the next section we discuss such a case, namely when A is a circular-ones matrix.

3. Circular-ones matrices. Call a zero-one matrix A a *circular-ones matrix* if in each row of A the ones occur in circular consecutive order. That is, in each row the ones or the zeros form a contiguous block. Closely related is the *circular-ones property* (see [15]) for matrices. A matrix has the circular-ones property if it can be transformed into a circular-ones matrix by permuting the columns. If it exits, such a permutation can be found in linear time (see [5]). If A is an $m \times n$ circular-ones matrix, then replacing each row a^{T} of A in which the ones do not form a contiguous block by $(\mathbf{1} - a)^{\mathsf{T}}$, we obtain an interval matrix, which is totally unimodular. Hence every circular-ones matrix is nearly totally unimodular. In this section we give an efficient algorithm for finding decompositions as in Theorem 1 in the special case of circular-ones matrices.

It will be convenient to use the following notation. For integers $i \leq j$, we denote the set $\{i, i+1, \ldots, j\}$ by $[i, j]$. For finite sets $U \subseteq V$ and $x \in \mathbb{R}^V$, we denote the characteristic vector of U by χ^U and define $x(U) := x^{\mathsf{T}}\chi^U$.

If $P = \{x \mid Ax \leq b,\ \mathbf{1}^{\mathsf{T}}x = s\}$, where A is an interval matrix, and b and s are integer, decomposing an integer vector $x \in rP$ into r integer vectors in P can be done in polynomial time. In fact, such a decomposition can be found that does not depend on the matrix A or the vector b. In the case that x is the characteristic vector of a subset $X \subseteq \{1, \ldots, n\}$, the decomposition simply amounts to coloring the ith element of X with color i modulo r. The proposed decomposition algorithm in Proposition 2 is not strongly polynomial, as it performs integer division on the coefficients of x. We will denote by size(x) the encoding length of a given vector $x \in \mathbb{Z}^n$.

Proposition 2. *Let integers r, s $(r > 0)$, and a vector $x \in \mathbb{Z}^n$ satisfying $\mathbf{1}^{\mathsf{T}}x = rs$ be given. Then we can find in time $O(n^2 + size(x))$ a decomposition*

$$x = \sum_{t=1}^{l} n_t x_t \tag{10}$$

of x into integer vectors x_t with $\mathbf{1}^{\mathsf{T}}x_t = s$ and such that for any interval $I \subseteq \{1, \ldots, n\}$ and any integer d we have $x(I) \leq rd \Rightarrow x_t(I) \leq d$ and $x(I) \geq rd \Rightarrow x_t(I) \geq d$, for each $t = 1, \ldots, l$. The numbers x_t are positive integers with $n_1 + \cdots + n_l = r$ and $l \leq n + 1$.

Proof. Define for $i = 1, 2, \ldots, n$ the integers z_i, q_i, and r_i by

$$\begin{aligned} z_i &:= x([1, i]), \\ z_i &= q_i r + r_i, \quad \text{where } 0 \leq r_i \leq r - 1. \end{aligned} \tag{11}$$

Sort the elements of the set $\{0, r\} \cup \{r_1, r_2, \ldots, r_n\}$ in increasing order to obtain $0 = r'_0 < r'_1 < \cdots < r'_l = r$. Now we define for $t = 1, 2, \ldots, l$ the numbers $n_t \in \mathbb{Z}_+$ and vectors $x_t \in \mathbb{Z}^n$ by

$$\begin{aligned} n_t &:= r'_t - r'_{t-1}, \\ x_t([1, i]) &:= q_i + \delta_{r_i \geq r'_t} \quad \text{for } i = 1, \ldots, n. \end{aligned} \tag{12}$$

Here δ denotes the Kronecker delta attaining the value 1 if the subscript is true and the value 0 if it is false. It is an easy verification that the numbers q_i and r_i can be found in time $O(n + \text{size}(x_1) + \cdots + \text{size}(x_n))$. Hence the x_t and n_t can be found in time $O(n^2 + \text{size}(x))$.

Clearly $l \leq n+1$, $n_1+\cdots+n_l = r$, and $\mathbf{1}^\mathsf{T} x_t = s$ for each t. Since for $i = 1, \ldots, n$

$$\begin{aligned}
(13) \qquad \sum_{t=1}^{l} n_t x_t([1,i]) &= \sum_{t=1}^{l} (r'_t - r'_{t-1})(q_i + \delta_{r_i \geq r'_t}),\\
&= rq_i + \sum_{t=1}^{l} (r'_t - r'_{t-1}) \delta_{r_i \geq r'_t},\\
&= rq_i + r_i,\\
&= x([1,i]),
\end{aligned}$$

we have that $x = n_1 x_1 + \cdots + n_l x_l$. For any t, t' and any interval $[i,j]$ we have (defining $q_0 := r_0 := 0$)

$$\begin{aligned}
(14) \qquad &|x_t([i,j]) - x_{t'}([i,j])|\\
&= |q_j - q_{i-1} + \delta_{r_j \geq r'_t} - \delta_{r_{i-1} \geq r'_t} - q_j + q_{i-1} - \delta_{r_j \geq r'_{t'}} + \delta_{r_{i-1} \geq r'_{t'}}|\\
&= |\delta_{r_j \geq r'_t} - \delta_{r_{i-1} \geq r'_t} - \delta_{r_j \geq r'_{t'}} + \delta_{r_{i-1} \geq r'_{t'}}| \leq 1.
\end{aligned}$$

This implies the proposition. □

Theorem 2. *Given an $m \times n$ matrix A which is (up to multiplying some rows by -1) a circular-ones matrix, vectors $b \in \mathbb{Z}^m$, $x \in \mathbb{Z}^n$, and a nonnegative integer k, we can test in time $O(n(n+m))$ whether $x \in kP_{A,b}$, and find in time $O(n(n+m)+size(x))$ a decomposition*

$$(15) \qquad x = \sum_{i=1}^{l} n_i x_i,$$

where x_i is an integer vector in $P_{A,b}$ and n_i is a positive integer for $i = 1, \ldots, l$; the positive numbers n_i satisfy $n_1 + \cdots + n_l = k$ and $l \leq 2n+2$.

Proof. Let $a = \mathbf{1}$ be the all-one vector of length n and let $c \in \{-1,0,1\}^m$ be defined by $c_i = A_{i,n}$. Let $\tilde{A} := A - ca^\mathsf{T}$. Then in each row of $\tilde{A}$ the nonzero elements form a contiguous block of either ones or minus ones. Write $a^\mathsf{T} x = qk + r$ as in Proposition 1. Now each of the inequalities in system (8) is of the form $y([i,j]) \leq d$ or $y([i,j]) \geq d$ for some integers $i, j \in \{1, \ldots, n\}$ and d. This implies that (8) can be solved by a shortest path algorithm as follows (see [10]). Construct a directed graph D with vertex set $\{0,1,2,\ldots,n\}$ and an arc $(i-1,j)$ of length d for each inequality $y([i,j]) \leq d$, and an arc $(j, i-1)$ of length $-d$ for each inequality $y([i,j]) \geq d$. Now the system is feasible if and only if D has no negative-length cycles. If there are no negative-length cycles, let d_i be the length of a shortest-length path starting in vertex i for $i = 0,1,2,\ldots,n$. Then the vector $y \in \mathbb{Z}^n$ defined by $y_i := d_i - d_{i-1}$ is an integer solution to (8). Detecting a negative-length cycle and finding the numbers d_i can be done in time $O(n(n+m))$ by the Bellman–Ford method (see [4]). If an integer solution y is found, we can by Proposition 2 find decompositions of y and $x - y$ into integer vectors in $P_{A,b}$ in time $O(n(n+m) + \text{size}(x))$. □

4. Applications.

In this section we discuss some applications of Theorems 1 and 2.

4.1. Proper circular arc graphs.

For two points a, b on the unit circle, the closed segment running clockwise from a to b is called an *arc* and is denoted by $[a,b]$. A *proper* circular arc system is a finite set of arcs $A_i := [a_i, b_i]$, $i = 1, \ldots, n$, with the property that $A_i \not\subseteq A_j$ for any two distinct $i, j \in \{1, \ldots, n\}$. The *proper circular arc graph* G associated with this system is the graph with vertex set $\{1, \ldots, n\}$, and two vertices i and j are joined by an edge if $A_i \cap A_j \neq \emptyset$. We will assume that the arcs are numbered in such a way that the points $a_1, a_2, \ldots, a_n$ occur in clockwise order around the circle. For each $i = 1, \ldots, n$, let $C_i := \{j \in \{1, \ldots, n\} \mid a_i \in A_j\}$. Note that because no arc contains another arc, the ones in the characteristic vector of C_i occur in circular consecutive order. Let A be the $n \times n$ matrix with the characteristic vectors of the sets C_i as rows. Then A is a circular-ones matrix, and $P_{\left(\begin{smallmatrix} A \\ -I \end{smallmatrix}\right), \left(\begin{smallmatrix} \mathbf{1} \\ \mathbf{0} \end{smallmatrix}\right)}$ is the stable set polytope of G. A k-coloring of G corresponds to a decomposition of the all-one vector into k integer vectors in the stable set polytope. As a corollary to Theorem 1, we find that the stable set polytope of a proper circular arc graph has the integer decomposition property. This result was proved by Niessen and Kind in [9]. By Theorem 2 we can find a coloring of G using a minimum number of colors in time $O(n^2 \log n)$ by binary search on the number of colors k. This is a result of Orlin, Bonucelli, and Bovet (see [10]).

4.2. A packet scheduling problem for smart antennas. In recent years, there has been a growing interest in adaptive antenna arrays known as “smart antennas,” for example, for use in third generation mobile telecommunication systems (see [11, 14]). A smart antenna may be viewed as a collection of colocated directive antennas in the plane that each transmit to (or receive from) a narrow beam (approximately 12 degrees). Each of these directive antennas can be independently oriented and can serve one user at a time. In order to avoid unwanted interference, there is a combinatorial restriction on the sets of users that can be served simultaneously: a user that is being served cannot be in the beam of a directive antenna that serves another user. This restricts the number of users that can be served during the same time slot. As an example, suppose that the angle of the beams from the directive antennas is 12 degrees and that three users are in a common sector of 12 degrees. If the middle of the three users is served, then the beam corresponding to the antenna that serves it must either contain the clockwise or the counterclockwise neighbor, which therefore cannot be served at the same time. This implies that for a set of users that are served simultaneously, the angle between any of these users and its second clockwise neighbor is more than 12 degrees. Hence the number of users that can be served in a single time slot is less than 60. In fact, we may assume that the number of available directive antennas is unlimited and that the sets of users that can be served simultaneously are determined exactly by this interference constraint.

In [1, 2] Amaldi, Capone, and Malucelli considered the following two scheduling problems. Associate with each user a number representing its priority. The first scheduling problem is to find a set of users that can be served in a single time slot, maximizing the sum of their priority numbers. In [2] they gave a polynomial-time algorithm for this scheduling problem by reducing it to the problem of finding a maximum weight directed path in a weighted acyclic digraph. Here we will focus on the second scheduling problem.

> Given a set of users, find a schedule for serving all the users that needs a minimum number of time slots. That is, give a partition of the users into a minimal number of classes, where the users in each class can be served simultaneously.

Amaldi, Capone, and Malucelli devised heuristics for this problem and asked whether the problem is NP-hard. As an application of Proposition 2 we will give an efficient algorithm for solving this packet scheduling problem.

In the scheduling problem, the exact positions of the users are not needed, only their direction as seen from the smart antenna. Hence we can model the users by points on the unit circle and let the beams from the directive antennas correspond to arcs of a fixed length α of the unit circle. Following [2], the scheduling problem can be formalized as follows.

Let $\alpha \in (0, 2\pi)$ be given. A finite set S of points on the unit circle will be called *independent*[1] if there exist $|S|$ arcs on the unit circle of length $\alpha > 0$ such that each point in S is in exactly one of these arcs and each of these arcs contains exactly one element of S. Note that any two of the $|S|$ arcs may intersect as long as the intersection does not contain a point in S. The independent sets correspond to the sets of users that can be served simultaneously. Given α and a finite subset V of the unit circle (the users), the packet scheduling problem can now be restated as follows.

Partition problem. *Given a finite subset V of the unit circle and an $\alpha > 0$, find a partition of V into a minimal number of independent sets.*

We will now show the connection between the partition problem and the circular-ones matrices. We make the following observation.

Observation. A finite set S of points on the unit circle is independent if and only if $|S \cap [s, s']| \leq 2$ for each arc $[s, s']$ of length α starting at a point $s \in S$.

Proof. To see necessity, suppose that some arc of length α contains $u, v, w \in S$ in this order; then any arc of length α containing v also contains u or w, and hence S is not independent. For sufficiency, suppose that $|S \cap [s, s']| \leq 2$ for each arc $[s, s']$ of length α with $s \in S$. Let $v \in S$ and let u and w be the counterclockwise and clockwise neighbor in S of v, respectively. The length of $[u, w]$ must be larger than α since $|[u, w] \cap S| > 2$, and hence there exists an arc of length α intersecting S only in v. □

Note that the last argument also shows that, given an independent set S, $|S|$ arcs of length α as in the definition of independent set are easily constructed from S. Now given a finite subset V of the unit circle, define for each $v \in V$ the row vector $a_v \in \{0, 1\}^V$ as the incidence vector of the intersection of V with the arc $[v, v']$ of length α. Let $Ax \leq b$ be the system consisting of the inequalities $a_v x \leq 2$ for $v \in V$, and the inequalities $\mathbf{0} \leq x \leq \mathbf{1}$. Then the matrix A is (up to signs of the rows) a circular-ones matrix, and the incidence vectors of the independent sets are precisely the integer vectors in $P_{A,b}$. Hence, the partition problem is to find a decomposition of the all-one vector into a minimal number of integer vectors in $P_{A,b}$. By Theorem 2 we can test whether V can be partitioned into k independent sets in time $O(n^2)$. Hence using binary search on k, we obtain an $O(n^2 \log n)$ algorithm for solving the packet scheduling problem.

4.3. Edge coloring nearly bipartite graphs. A graph G is called *nearly bipartite* if we can obtain a bipartite graph by deleting a vertex from G. Let $G = (V, E)$ be a nearly bipartite graph and let $u \in V$ be a vertex of G such that $G - u$ is bipartite with bipartition $V \setminus \{u\} = V_1 \cup V_2$. Let A be the $V \times E$ incidence matrix of G. Then A is nearly totally unimodular. Indeed, let $a := \chi^F$, where $F \subset E$ is the set of edges between u and V_2, and define $\tilde{A} := A - \chi^{\{u\}} a^\mathsf{T}$. Then $\binom{\tilde{A}}{a^\mathsf{T}}$ is the incidence matrix of a bipartite graph G' obtained from G by splitting u into two points. As the incidence matrix of a bipartite graph is totally unimodular, this implies that A is nearly totally unimodular. It now follows from Theorem 1 that the matching polytope $P_{\binom{A}{-I}, \binom{\mathbf{1}}{\mathbf{0}}}$ of

[1] In [2] arcs are half-open segments, but for the definition, this is equivalent to using closed segments of the same length.

G has the integer decomposition property. Equivalently, the chromatic index of G is the roundup of the fractional chromatic index. This result was proved in [7] and [12].

If G is viewed as a multigraph by taking each edge e with multiplicity $x_e \in \mathbb{N}$, then finding a k-edge coloring G with k minimal can be done as follows. First observe that we may assume that $k \geq \Delta$, where Δ is the maximum degree of a vertex in G (counting multiplicities). Hence we have $q = 0$ in Theorem 1, and $r = x(F)$ is the number of edges from u to V_2. Solving system (4) amounts to finding an integer vector y satisfying $\mathbf{0} \leq y \leq x$ and

$$\begin{aligned} y(e) &= 0 \quad \text{for } e \in \delta(u) \setminus F, \\ y(e) &= x(e) \quad \text{for } e \in F, \\ y(\delta(v)) &\leq x(F) \quad \text{for } v \in V \setminus \{u\}, \\ y(\delta(v)) &\geq x(F) + x(\delta(v)) - k \quad \text{for } v \in V \setminus \{u\}. \end{aligned} \tag{16}$$

This can be done by reducing it to a flow problem with capacities and demands on the arcs. Hence we can find an integer solution y (if it exists) in time $O(mn \log n)$ (see [13]). Since for $k \geq \Delta + x(F)$ we may take $y = \chi^F$, to find the minimal k we need to check at most $O(\log x(F))$ values of k using binary search. If an integer solution y is found, decomposing y and $x - y$ as in Theorem 1 comes down to capacitated edge coloring of the bipartite graphs $(V, E \setminus (\delta(u) \setminus F))$ and $(V, E \setminus F)$, respectively. This can be done in time $O(m^2)$ (see [13]).

When x is a zero-one vector, the above algorithm comes down to the edge coloring algorithm as presented in [12].

Acknowledgments. The author would like to thank the anonymous referees for their many valuable suggestions, in particular for shortening the proof of the implication (i) ⇒ (ii) in Theorem 1.

REFERENCES

[1] E. Amaldi, A. Capone, and F. Malucelli, *Circular arc models and algorithms for packet scheduling in smart antennas*, in Fourth ALIO/EURO Workshop on Applied Combinatorial Optimization, Pucón, Chile, 2002.

[2] E. Amaldi, A. Capone, and F. Malucelli, *Discrete models and algorithms for packet scheduling in smart antennas*, Proceedings of the Second Cologne-Twente Workshop on Graphs and Combinatorial Optimization, Electron. Notes Discrete Math. 13, Elsevier, Amsterdam, 2003, pp. 1–4.

[3] S. Baum and L. E. Trotter, Jr., *Integer rounding and polyhedral decomposition for totally unimodular systems*, in Optimization and Operations Research (Proc. Workshop, Univ. Bonn, Bonn 1977; R. Henn, B. Korte and W. Oettli, eds.), Lecture Notes in Econom. Math. Systems 157, Springer, Berlin, 1978, pp. 15–23.

[4] R. E. Bellman, *On a routing problem*, Quart. Appl. Math., 16 (1958), pp. 87–90.

[5] K. S. Booth and G. S. Lueker, *Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms*, J. Comput. System Sci., 13 (1976), pp. 335–379.

[6] W. Cook, J. Fonlupt, and A. Schrijver, *An integer analogue of Carathéodory's theorem*, J. Combin. Theory Ser. B, 40 (1986), pp. 63–70.

[7] L. Eggan and M. Plantholt, *The chromatic index of nearly bipartite multigraphs*, J. Combin. Theory Ser. B, 40 (1986), pp. 71–80.

[8] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, 2nd ed., Springer-Verlag, Berlin, 1988.

[9] T. Niessen and J. Kind, *The round-up property of the fractional chromatic number for proper circular arc graphs*, J. Graph Theory, 33 (2000), pp. 256–267.

[10] J. B. Orlin, M. A. Bonuccelli, and D. P. Bovet, *An $O(n^2)$ algorithm for coloring proper circular arc graphs*, SIAM J. Algebraic Discrete Methods, 2 (1981), pp. 88–93.

[11] A. Perez-Neira, X. Mestre, and J. R. Fonollosa, *Smart antennas in software radio base stations*, IEEE Communications Magazine, 39 (2001), pp. 166–173.

[12] B. Reed, *Edge coloring nearly bipartite graphs*, Oper. Res. Lett., 24 (1999), pp. 11–14.

[13] A. Schrijver, *Combinatorial Optimization. Polyhedra and Efficiency. Vol. A*, Springer-Verlag, Berlin, 2003.

[14] K. Sheikh, D. Gesbert, D. Gore, and A. Paulraj, *Smart antennas for broadband wireless access networks*, IEEE Communications Magazine, 37 (1999), pp. 100–105.

[15] A. Tucker, *Matrix characterizations of circular-arc graphs*, Pacific J. Math., 39 (1971), pp. 535–545.

SIAM J. DISCRETE MATH.
Vol. 19, No. 3, pp. 807–813

A GENUS BOUND FOR DIGITAL IMAGE BOUNDARIES*

LOWELL ABRAMS† AND DONNIELL E. FISHKIND‡

Abstract. Shattuck and Leahy [*IEEE Trans. Med. Imag.*, 20 (2001), pp. 1167–1177] conjectured—and Abrams, Fishkind, and Priebe [*IEEE Trans. Med. Imag.*, 21 (2002), pp. 1564–1566], [*IEEE Trans. Med. Imag.*, 23 (2004), pp. 655–657] proved—that the boundary of a digital image is topologically equivalent to a sphere if and only if certain related foreground and background graphs are both trees. In this article we extend this result by proving upper and lower bounds on digital image boundary genus in terms of the foreground and background graphs, and we show that these bounds are best possible. Our results have current application to topology correction in medical imaging.

Key words. digital image, digital topology, combinatorial topology, surface

AMS subject classifications. 05C10, 57M15

DOI. 10.1137/S0895480104445691

1. Overview.

Digital topology is an area of great theoretical interest, having the additional bonus of significant application in imaging science and related areas. Our results are mathematical—the notation and setting are detailed in section 2—but we begin with a brief description of a current application.

The human cerebral cortex, when viewed as closed at the brain stem, is topologically like a sphere. Magnetic resonance imaging (MRI) can differentiate between tissue that is interior to the cerebral cortex and tissue that is exterior to the cerebral cortex. Because of the finiteness of resolution, what is generated by MRI is a three-dimensional array of cubes, each cube classified by MRI as "foreground" (tissue interior to the cerebral cortex) or "background" (tissue exterior to the cerebral cortex), and the boundary between the foreground and background is an approximation of the cerebral cortex itself.

Although topologically spherical, the cerebral cortex is densely folded, and the finite resolution, as well as noise, may lead to topological "handles" that do not actually exist. The physiological and neurological function of regions of the cerebral cortex, as well as the relationship between the regions, is dictated by the spherical topology rather than just spatial proximity. It is therefore important to "correct" the topology, and a number of different strategies are currently used [3], [4].

The strategy of Shattuck and Leahy [4] is fundamentally based on the construction of certain foreground and background graphs related to the MRI data; they conjectured that the image boundary is topologically spherical if and only if both foreground and background graphs are trees. In situations where one or both of the graphs are not trees, the edges are weighted to reflect corresponding junctional thickness, and a maximum weight spanning tree is found. Edges not on the spanning tree are removed by adjusting the image at corresponding locations, and the resulting image is then, by their conjecture, topologically spherical.

[*] Received by the editors August 17, 2004; accepted for publication (in revised form) March 18, 2005; published electronically December 7, 2005.
http://www.siam.org/journals/sidma/19-3/44569.html

[†] Department of Mathematics, The George Washington University, Washington, DC 20052 (labrams@gwu.edu). This author's research was partially supported by NSA grant MDA904-03-1-0023.

[‡] Department of Applied Mathematics and Statistics, The Johns Hopkins University, Baltimore, MD 21218-2682 (fishkind@ams.jhu.edu).

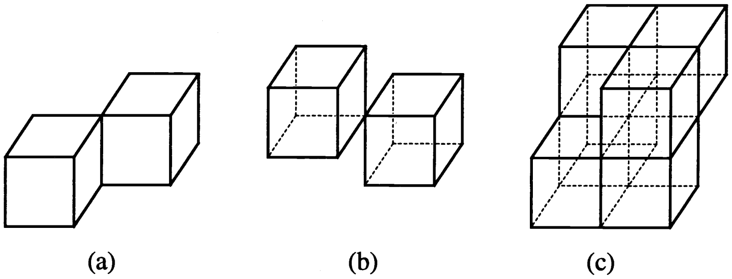


FIG. 1. *Voxel configurations which are forbidden if $\partial\mathcal{I}$ is a surface.*

The Shattuck and Leahy conjecture was proven and generalized by Abrams, Fishkind, and Priebe in [1] and [2], and the main result in this paper, Theorem 2, represents a further generalization. Theorem 2—articulated in section 2 and proven in section 3—gives bounds for the genus of the boundary of a digital image in terms of the foreground and background graphs, and these bounds are shown to be best possible. The truth of Shattuck and Leahy's conjecture is, in fact, a special case of our Theorem 2, and the bounds in Theorem 2 also provide the possibility of adapting Shattuck and Leahy's topology correction approach to imaged objects of higher genus.

2. Digital images, associated graphs, and the main result. A subset of $\mathbb{R}^3$ is a *surface* if it is compact, connected, and locally homeomorphic to an open disk in $\mathbb{R}^2$. Let N be a fixed positive integer. For any triplet of integers $(i,j,k) \in \{1,2,\dots,N\}^3$, define the *voxel* $v_{i,j,k}$ to be the closed Euclidean unit-cube $[i-\frac{1}{2},i+\frac{1}{2}] \times [j-\frac{1}{2},j+\frac{1}{2}] \times [k-\frac{1}{2},k+\frac{1}{2}]$. For any $A \subseteq \{1,2,\dots,N\}^3$, define the *digital image* $\mathcal{I}_A := \cup_{(i,j,k)\in A} v_{i,j,k}$. The digital image $\mathcal{I}_A$ is called *surrounded* if none of i, j, or k equals 1 or N, and $\mathcal{I}_A$ is called *standard* if $\mathcal{I}_A$ is surrounded and its boundary $\partial\mathcal{I}_A$ is a surface. The complementary digital image $\mathcal{I}_A^c$ is defined as $\mathcal{I}_{A^c}$, where $A^c := \{1,2,\dots,N\}^3\backslash A$. When there is no confusion we write $\mathcal{I}$ in place of $\mathcal{I}_A$.

For surrounded digital image $\mathcal{I}$, when is $\partial\mathcal{I}$ a surface? Of course, $\partial\mathcal{I}$ is always compact. It is shown in [2] that $\partial\mathcal{I}$ is locally homeomorphic to a disk if and only if it does not have any of the three "forbidden" voxel configurations[1] illustrated in Figure 1. When $\partial\mathcal{I}$ is locally homeomorphic to a disk it is not difficult to show that $\partial\mathcal{I}$ is connected, hence a surface, if and only if both $\mathcal{I}$ and $\mathcal{I}^c$ are connected.

For each $k \in \{1,2,\dots,N\}$, the *kth level* is $L_k := \cup_{i,j\in\{1,2,\dots,N\}} v_{i,j,k}$, and the $(k,k+1)$*th sheet* is $S_{k,k+1} := L_k \cap L_{k+1}$. Associated with a digital image $\mathcal{I}$ is the (multi)graph $G_\mathcal{I}$ with vertex set $V_\mathcal{I}$ and edge set $E_\mathcal{I}$ defined as follows: for each $k \in \{1,2,\dots,N\}$, we declare each connected component of $\mathcal{I} \bigcap L_k$ to be a vertex in $V_\mathcal{I}$. For any two vertices u and v on adjacent levels, say L_k and L_{k+1}, we declare each connected component of $u \cap v \subseteq S_{k,k+1}$ to be an edge in $E_\mathcal{I}$ whose graph-theoretic endpoints are u and v. When referring to a vertex $u \in V_\mathcal{I}$ or an edge $\epsilon \in E_\mathcal{I}$, context will dictate whether we are viewing u or ϵ as *Euclidean* subsets, i.e., subsets of $\mathbb{R}^3$, or as discrete graph-theoretic objects. In Figures 2 and 4 we show examples of $\mathcal{I}$, $G_\mathcal{I}$, and $G_{\mathcal{I}^c}$.

The following result was conjectured by Shattuck and Leahy [4] and proved by Abrams, Fishkind, and Priebe [1].

[1] In [2] we discuss a corrective strategy—involving slightly altering the digital image—for medical imaging applications in which $\partial\mathcal{I}$ is not locally homeomorphic to a disk.

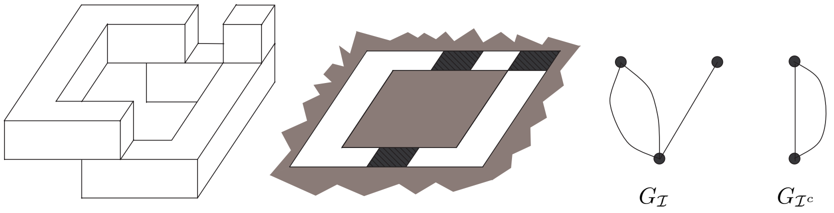


FIG. 2. *The drawing in the middle shows the intersection of the two levels on the left; the gray areas denote background edges and the black areas denote foreground edges.*

THEOREM 1 (spherical homeomorphism theorem). *For any standard digital image $\mathcal{I}$, $\partial\mathcal{I}$ is a topological sphere if and only if both $G_{\mathcal{I}}$ and $G_{\mathcal{I}^c}$ are graph-theoretic trees.*

For digital image $\mathcal{I}$, define $r_{\mathcal{I}} := |E_{\mathcal{I}}| - |V_{\mathcal{I}}| + 1$. This value is called the *corank* or *cycle rank* of $G_{\mathcal{I}}$ when $G_{\mathcal{I}}$ is connected. If $\mathcal{I}$ is connected, then $G_{\mathcal{I}}$ is connected as well, so $r_{\mathcal{I}} \geq 0$. In particular, $r_{\mathcal{I}} = 0$ if and only if $G_{\mathcal{I}}$ is a tree. The cycle ranks $r_{\mathcal{I}}$ and $r_{\mathcal{I}^c}$ are the first Betti numbers of $G_{\mathcal{I}}$ and $G_{\mathcal{I}^c}$, respectively. The following is our main result; $g(\partial\mathcal{I})$ denotes the genus of the surface $\partial\mathcal{I}$, which is the first Betti number of $\mathcal{I}$.

THEOREM 2. *For any standard digital image $\mathcal{I}$,*

$$\max\{r_{\mathcal{I}}, r_{\mathcal{I}^c}\} \leq \; g(\partial\mathcal{I}) \leq \; r_{\mathcal{I}} + r_{\mathcal{I}^c}.$$

Moreover, this is best possible in the sense that, for any nonnegative integers a, b, and c such that $\max\{a, b\} \leq c \leq a + b$, there exists a standard digital image $\mathcal{I}$ such that $r_{\mathcal{I}} = a$, $r_{\mathcal{I}^c} = b$, and $g(\partial\mathcal{I}) = c$.

It is not hard to see that Theorem 2 implies Theorem 1: for standard digital image $\mathcal{I}$, if $\partial\mathcal{I}$ is topologically spherical, then $\max\{r_{\mathcal{I}}, r_{\mathcal{I}^c}\} \leq 0$ implies that both $G_{\mathcal{I}}$ and $G_{\mathcal{I}^c}$ are trees, and conversely, if both $G_{\mathcal{I}}$ and $G_{\mathcal{I}^c}$ are trees, then $g(\partial\mathcal{I}) \leq 0 + 0$ implies that $\partial\mathcal{I}$ is topologically spherical. We may therefore think of Theorem 2 as a generalization of the spherical homeomorphism theorem.

3. Proof of the main result, Theorem 2. We begin the proof of Theorem 2 with the following slightly weakened version.

LEMMA 3. *For any standard digital image $\mathcal{I}$, $r_{\mathcal{I}} \;\leq\; g(\partial\mathcal{I}) \;\leq\; r_{\mathcal{I}} + r_{\mathcal{I}^c}$.*

The proof of Lemma 3 extends and refines the development and strategies in [1].

If a surface S has a 2-cell embedding of some graph H with n vertices, e edges, and f faces, then Euler's classical result states that $n - e + f = 2 - 2g(S)$. The value $\chi(S) \overset{\text{def}}{=} n - e + f$ is called the *Euler characteristic* of S. Our assumption that $\partial\mathcal{I}$ is a surface implies that, for every $v \in V_{\mathcal{I}}$, ∂v is a surface; it is useful to view the voxel vertices, voxel edges, and voxel faces on $\partial\mathcal{I}$ or ∂v, respectively, as a 2-cell embedding of a graph on $\partial\mathcal{I}$ or ∂v.

Suppose $\epsilon \in E_{\mathcal{I}}$ is a subset of sheet S; the assumption that $\partial\mathcal{I}$ is a surface implies that the boundary of ϵ in S, denoted $\partial_S\epsilon$, consists of a disjoint union of simple, closed curves. Let h_e denote the number of "punctures" in ϵ; i.e., h_ϵ is one less than the number of connected components of $\partial_S\epsilon$. Even though ϵ and $\partial_S\epsilon$ are not surfaces (they have boundaries, and $\partial_S\epsilon$ may not be connected) the Euler characteristics $\chi(\epsilon)$ and $\chi(\partial_S\epsilon)$ are well defined; in fact, $\chi(\partial_S\epsilon) = 0$ (since it has equal numbers of voxel edges and voxel vertices, and no faces), and

$$\chi(\epsilon) = 2 - (h_\epsilon + 1) = 1 - h_\epsilon. \tag{1}$$

Proof of Lemma 3. Since the relative interior of each $\epsilon \in E_{\mathcal{I}}$ is a subset of two vertex boundaries not contained in $\partial\mathcal{I}$, and the relative boundary of each ϵ has Euler characteristic 0, a simple inclusion-exclusion argument gives

$$\chi(\partial\mathcal{I}) = \sum_{v\in V_{\mathcal{I}}} \chi(\partial v) - 2\sum_{\epsilon\in E_{\mathcal{I}}} \chi(\epsilon). \tag{2}$$

Next, note that there is a natural one-to-one correspondence between the genus holes[2] of the vertices of $V_{\mathcal{I}}$ and the vertices of $V_{\mathcal{I}^c}$ other than the N "outermost" vertices of $V_{\mathcal{I}^c}$, one per level. Thus

$$\sum_{v\in V_{\mathcal{I}}} \chi(\partial v) = \sum_{v\in V_{\mathcal{I}}} [2-2g(\partial v)] = 2|V_{\mathcal{I}}| - 2(|V_{\mathcal{I}^c}| - N). \tag{3}$$

Combining (1), (2), and (3), we obtain

$$\begin{aligned} 2-2g(\partial\mathcal{I}) \;=\; \chi(\partial\mathcal{I}) &= \sum_{v\in V_{\mathcal{I}}} \chi(\partial v) - 2\sum_{\epsilon\in E_{\mathcal{I}}} \chi(\epsilon) \\ &= 2|V_{\mathcal{I}}| - 2(|V_{\mathcal{I}^c}| - N) - 2\sum_{\epsilon\in E_{\mathcal{I}}} (1-h_\epsilon) \\ &= 2 - 2\left(|E_{\mathcal{I}}| - |V_{\mathcal{I}}| + 1\right) - 2\left(|V_{\mathcal{I}^c}| - N - \sum_{\epsilon\in E_{\mathcal{I}}} h_\epsilon\right), \end{aligned}$$

and it follows that

$$g(\partial\mathcal{I}) \;=\; r_{\mathcal{I}} \;+\; \left(|V_{\mathcal{I}^c}| - N - \sum_{\epsilon\in E_{\mathcal{I}}} h_\epsilon\right). \tag{4}$$

For $k = 1, 2, \ldots, N-1$, let $b_{k,k+1}$ denote the number of connected components in the subgraph of $G_{\mathcal{I}^c}$ induced by the vertices of $V_{\mathcal{I}^c}$ in the kth and $(k+1)$th levels, and denote by $B_{k,k+1}$ the Euclidean set $S_{k,k+1}\backslash\bigcup_{\epsilon\in E_{\mathcal{I}}:\epsilon\subseteq S_{k,k+1}} \epsilon$. Observe that the Euclidean set $B_{k,k+1}$ has $b_{k,k+1}$ components.

It now follows that

$$1 + \sum_{\epsilon\in E_{\mathcal{I}}:\epsilon\subseteq S_{k,k+1}} h_\epsilon \;=\; b_{k,k+1}.$$

Summing this equation over k yields

$$N - 1 + \sum_{\epsilon\in E_{\mathcal{I}}} h_\epsilon \;=\; \sum_{k=1}^{N-1} b_{k,k+1}. \tag{5}$$

Substituting (5) into (4) and simplifying, we find that Lemma 3 is now equivalent to the assertion that, for any standard digital image $\mathcal{I}$,

$$2\left(|V_{\mathcal{I}^c}| - 1\right) - |E_{\mathcal{I}^c}| \;\leq\; \sum_{k=1}^{N-1} b_{k,k+1} \;\leq\; |V_{\mathcal{I}^c}| - 1. \tag{6}$$

[2] Suppose $v \in V_{\mathcal{I}}$ is in level L. We use the term "genus hole" of v to refer to each component of the complement of v in L which is bounded horizontally in $\mathbb{R}^3$ by v. For each genus hole of v, there is a unique $w \in V_{\mathcal{I}^c}$ such that this genus hole of v is precisely the union of w and w's genus holes.

To show the right-hand side of (6), suppose first that we remove all edges of $G_{\mathcal{I}^c}$. Without any edges, $\sum_{k=1}^{N-1} b_{k,k+1} = 2|V_{\mathcal{I}^c}| - 2$, since each vertex in $V_{\mathcal{I}^c}$, with the exception of the single vertices in L_1 and L_N, respectively, is counted as a distinct component in the tabulation of $b_{k-1,k}$ as well as $b_{k,k+1}$ for some k (the single vertices in L_1 and L_N are tabulated only once, in $b_{1,2}$ and $b_{N-1,N}$, respectively). Now consider a spanning tree T of the original graph $G_{\mathcal{I}^c}$. Each of the $|V_{\mathcal{I}^c}| - 1$ edges of T, when returned to $G_{\mathcal{I}^c}$, reduces the total number of components by 1, so $\sum_{k=1}^{N-1} b_{k,k+1} = |V_{\mathcal{I}^c}| - 1$ after all the edges of T have been restored to $G_{\mathcal{I}^c}$. Returning the remaining edges to the original graph $G_{\mathcal{I}^c}$ can only further reduce the sum, and thus the right-hand side of (6) holds. Since reduction of the sum occurs only through this returning of edges, the largest possible reduction equals $|E_{\mathcal{I}^c}|$, confirming the left-hand side of (6) and completing the proof of Lemma 3. □

LEMMA 4. *For any standard digital image $\mathcal{I}$, $r_{\mathcal{I}^c} \leq g(\partial\mathcal{I})$.*

The proof of Lemma 4, together with Lemma 3, will complete the proof of the bound $\max\{r_{\mathcal{I}}, r_{\mathcal{I}^c}\} \leq g(\partial\mathcal{I}) \leq r_{\mathcal{I}} + r_{\mathcal{I}^c}$ in Theorem 2. However, using the approach of the proof of Lemma 3 to show $r_{\mathcal{I}^c} \leq g(\partial\mathcal{I})$ requires more than simply reversing the roles of $\mathcal{I}$ and $\mathcal{I}^c$ since $\mathcal{I}^c$, unlike $\mathcal{I}$, is not standard.

Proof of Lemma 4. Enlarge the ambient space for digital images to allow voxels with one or more coordinates equal to 0 or $N+1$. Accordingly, use the adjective *surrounded* to indicate that a digital image contains no voxel with any coordinate equal to 0 or $N+1$; note that any digital image which is standard in the previous sense remains standard when "surrounded" is redefined in this way.

For the sake of simplifying notation, we let $\mathcal{I}$ denote a standard digital image which is surrounded in the original, smaller, ambient space, and let $\mathcal{I}^c$ denote its complement in that smaller space. Let $v_{i'j'k'}$ denote a voxel in $\mathcal{I}$ with minimum k'. Let $\mathcal{J}$ denote the digital image consisting of the union of the following Euclidean sets: $\mathcal{I}$, the voxels v_{ijk} such that at least one of i, j, k is 0 or $N+1$, and the voxels $v_{i'j'k}$ such that $0 \leq k < k'$. Note that $\mathcal{J}$ and $\mathcal{J}^c$ are connected, and $\mathcal{J}^c$ is standard. Moreover, we have topological equivalence of $\partial\mathcal{I}$ and $\partial\mathcal{J}^c$, since the change in $\partial\mathcal{I}$ amounts to cutting out an open disk, attaching one end of a tube to $\partial\mathcal{I}$ along the boundary of the removed disk, and then capping off the tube. (We consider $\partial\mathcal{J}^c$ rather than $\partial\mathcal{J}$ because $\partial\mathcal{J}$ contains the additional component $\partial(\mathcal{J} \cup \mathcal{J}^c)$.) Figure 3 illustrates this process.

For each k in $1, \ldots, k'-1$ there is exactly one vertex of $G_{\mathcal{I}^c}$ in level k, and no vertices of $G_{\mathcal{I}}$. Since the removal of the voxels $\{v_{i'j'k} \mid 0 \leq k < k'\}$ does not disconnect any of the vertices of $G_{\mathcal{I}^c}$, and no other changes are made to $\mathcal{I}^c$ in the process of constructing $\mathcal{J}^c$, we see that $G_{\mathcal{J}^c} = G_{\mathcal{I}^c}$, and thus $r_{\mathcal{J}^c} = r_{\mathcal{I}^c}$.

Applying Lemma 3 to the standard digital image $\mathcal{J}^c$ yields

$$r_{\mathcal{I}^c} \ = \ r_{\mathcal{J}^c} \ \leq \ g(\partial\mathcal{J}^c) \ = \ g(\partial\mathcal{I}),$$

and Lemma 4 is shown. □

It is interesting to note that there is also a close relationship between $G_{\mathcal{I}}$ and $G_{\mathcal{J}}$. The voxels $\{v_{ijk} \mid \text{at least one of } i,j,k \text{ is 0 or } N+1\} \cup \{v_{i'j'k} \mid 0 \leq k < k'\}$ give rise to a path P in graph $G_{\mathcal{J}}$. In fact, $G_{\mathcal{J}}$ can be obtained from $G_{\mathcal{I}}$ by attaching P at a single endpoint. Since the edges of P lie in no cycles, we have $r_{\mathcal{J}} = r_{\mathcal{I}}$.

We now establish that the bounds in Theorem 2 are "best possible."

LEMMA 5. *For any nonnegative integers a, b, and c such that* $\max\{a,b\} \leq c \leq a+b$, *there is a standard digital image $\mathcal{I}$ such that $r_{\mathcal{I}} = a$, $r_{\mathcal{I}^c} = b$, and $g(\partial\mathcal{I}) = c$.*

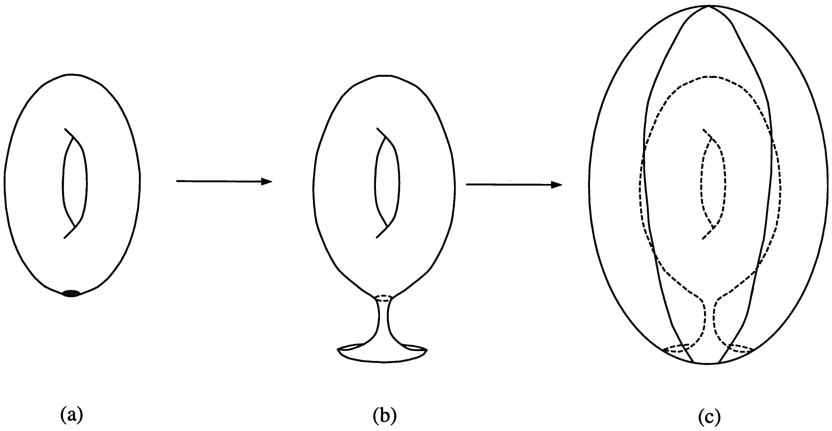


FIG. 3. *An illustration of the topological effect of modifying $\mathcal{I}$ to obtain $\mathcal{J}$:* (a) *$\partial\mathcal{I}$ with a disk cut out,* (b) *with a tube attached, and* (c) *with the tube capped off.*

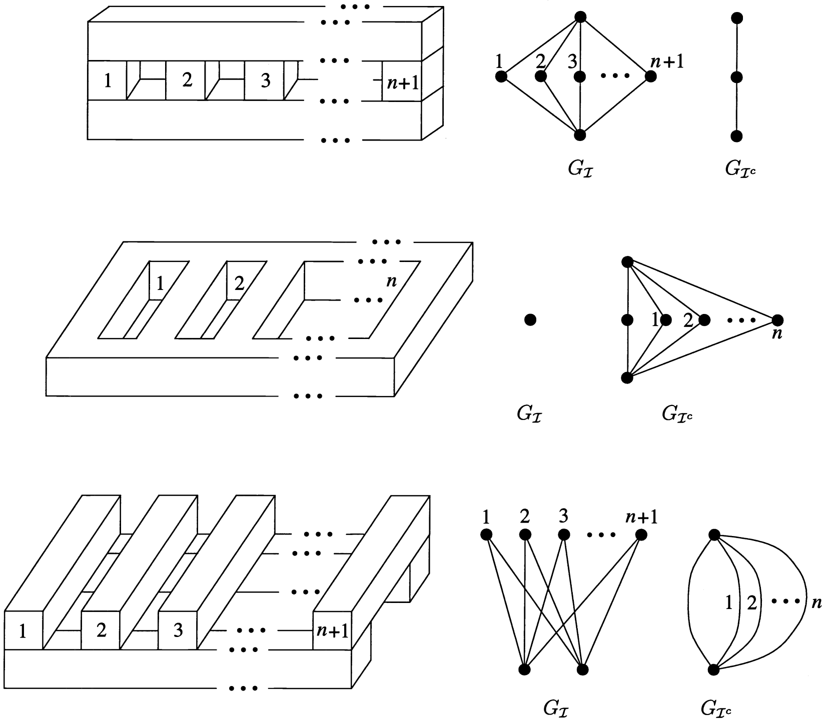


FIG. 4. *Three key examples—vertical n-torus, horizontal n-torus, and n-ladder—used in the proof of Theorem* 2*. Note that vertices of degree* 1 *corresponding to levels which contain no voxel in $\mathcal{I}$ have been omitted from the graphs $G_{\mathcal{I}^c}$.*

Proof of Lemma 5. Suppose a, b, and c are nonnegative integers such that $\max\{a, b\} \leq c \leq a+b$. Construct a digital image $\mathcal{I}$ by connecting, in any topologically trivial way, a vertical $(c-b)$-torus, a horizontal $(c-a)$-torus, and an $(a+b-c)$-ladder (see Figure 4). Note that $g(\partial\mathcal{I}) = (c-b) + (c-a) + (a+b-c) = c$, $r_{\mathcal{I}} = (c-b) + (a+b-c) = a$, and $r_{\mathcal{I}^c} = (c-a) + (a+b-c) = b$. $\square$

Theorem 2 now follows directly from Lemmas 3, 4, and 5. $\square$

Acknowledgment. We thank the anonymous referees for their thoughtful suggestions.

REFERENCES

[1] L. Abrams, D. E. Fishkind, and C. E. Priebe, *A proof of the spherical homeomorphism conjecture for surfaces*, IEEE Trans. Med. Imag., 21 (2002), pp. 1564–1566.

[2] L. Abrams, D. E. Fishkind, and C. E. Priebe, *The generalized spherical homeomorphism theorem for digital images*, IEEE Trans. Med. Imag., 23 (2004), pp. 655–657.

[3] X. Han, C. Xu, U. Braga-Neto, and J. Prince, *Topology correction in brain cortex segmentation using a multiscale, graph-based algorithm*, IEEE Trans. Med. Imag., 21 (2002), pp. 109–121.

[4] D. W. Shattuck and R. M. Leahy, *Automated graph-based analysis and correction of cortical volume topology*, IEEE Trans. Med. Imag., 20 (2001), pp. 1167–1177.

SIAM J. Discrete Math.
Vol. 19, No. 4, pp. 815–847

GRAPH MINORS AND RELIABLE SINGLE MESSAGE TRANSMISSION*

FAITH ELLEN FICH[†], ANDRÉ KÜNDGEN[‡], MICHAEL J. PELSMAJER[§], AND RADHIKA RAMAMURTHI[‡]

Abstract. End-to-end communication considers the problem of sending messages from a sender s to a receiver r through an asynchronous, unreliable network, such as the Internet. We consider the problem of transmitting a single message from s to r through a network in which edges may fail and cannot recover. We assume that some sr-path survives, but we do not know which path it is. We are concerned with protocols that do not store information at intermediate nodes and that ensure that a message sent by s will be recieved by r (no matter which edges fail) without generating an infinite number of messages.

We explicitly characterize the family of networks for which there is such a protocol using headerless packets. This characterization is given in terms of forbidden rooted minors, which leads to a linear time recognition algorithm for this family of networks. We obtain a similar characterization for the family of networks in which a message can be broadcast from a single vertex s to all other vertices. Finally, we show that there is a forbidden rooted minor characterization for the more general case when a header (containing routing information) of constant length is attached to the message, and we discuss the algorithmic consequences of this characterization.

Key words. end-to-end communication, reliable transmission, graph minors, tree-width, tree decomposition

AMS subject classifications. 68M10, 05C90, 05C83, 68R10, 05C85, 94C15

DOI. 10.1137/S0895480103421129

1. Introduction.

End-to-end communication considers the problem of sending messages from a sender s to a receiver r through an asynchronous, unreliable network. The network can be modeled by a bi-rooted graph where the vertices represent processors, the edges represent links between the processors, and the vertices s and r are the roots. Packets are sent along the links of the network. A packet contains a message and may contain a header with routing information. We generally seek a protocol which allows algorithms designed for reliable networks to be run on unreliable networks, such as the Internet.

We consider the simplified problem of transmitting a single message from s to r through a network in which edges may fail and cannot recover. We assume that packets traveling along an edge are received in the same order in which they are sent, but we make no other assumptions concerning the speed at which packets travel. In particular, an edge that has failed is indistinguishable from an edge along which packets are traveling very slowly. Therefore, a solution cannot use information about which edges have failed. In addition, we assume that there is no edge-cut of failed edges separating s and r; otherwise transmitting information from s to r is clearly impossible.

*Received by the editors January 13, 2003; accepted for publication (in revised form) May 1, 2005; published electronically December 30, 2005.

http://www.siam.org/journals/sidma/19-4/42112.html

[†]Department of Computer Science, University of Toronto, Toronto M55 1A4, Canada (fich@cs.toronto.edu).

[‡]California State University, San Marcos, San Marcos, CA 92096 (akundgen@csusm.edu).

[§]Department of Applied Mathematics, Illinois Institute of Technology, Chicago, IL 60616 (pelsmajer@.iit.edu).

We seek a protocol that ensures that a message sent by s will be received by r. For this it would suffice to have each vertex simply forward a copy of any arriving packet to all of its neighbors. However, a protocol is undesirable if too much packet traffic is generated. Therefore we restrict attention to *always finite protocols*, which terminate without generating an infinite number of packets, no matter what links of the network are operational. We say that a protocol *ensures correct delivery* of a message from s to r in a network, if the protocol is always finite and if a message sent by s will be received by r, provided there is a surviving path of links between s and r (see [1]).

In public networks, intermediate processors may store fixed information about the network (such as the number of vertices) but do not store information about the state of the communication between s and r. Instead, packets have headers which contain routing information, and the headers are used to control the communication. We will restrict our attention to *memoryless protocols*: when a processor receives a packet, it must decide to which of its neighbors to send packets and what their new headers will be, basing its decision only on fixed information about the network stored at the node. Since the message contained in the packet does not affect the behavior of the protocol, there is a clear separation between the protocol and the application programs that use the protocol.

It is desirable to use small packet headers. We say that a network with roots s and r allows *reliable single message transmission with headers of size* d if there exists a memoryless protocol using d-bit packet headers that ensures correct delivery of a single message from s to r. In the special case when $d = 0$ we also refer to this as *reliable headerless single message transmission*.

In section 4 we show that for every constant d, there is a forbidden rooted minor characterization for networks that allow reliable single message transmission with headers of size d. This implies the existence of a linear-time algorithm for recognizing such networks (see section 6). Theorem 8.1 gives the family of forbidden rooted minors that characterize networks that allow reliable headerless single message transmission. This yields a feasible linear time algorithm for recognizing such networks. In section 8 we give analogous results for *broadcast networks*, in which one processor is the sender and all others are receivers. The remainder of the paper is devoted to the proof of Theorem 8.1. The proof uses properties of 2- and 3-connected graphs as well as tree-decompositions.

2. Related work. The hop count protocol (see [13]) is a simple memoryless protocol that ensures correct delivery of a message. To do so, the sender s sends a packet with header 1 and the message to each of its neighbors. On receipt of a packet, each intermediate vertex forwards a copy of the message to each of its other neighbors with the header incremented by one, provided that the new header does not exceed the length of the longest path from s to r. Headers of $\lceil \log_2 n \rceil$ bits suffice for any n-vertex graph.

Adler and Fich [1] proved that the hop count algorithm is optimal. That is, for the complete n-vertex graph K_n, headers of size $\log_2 n - O(1)$ are necessary to ensure correct delivery of a message using a memoryless protocol. However, for certain families of graphs, it is possible to do substantially better. For example, graphs with a feedback vertex sets of size f allow memoryless protocols that ensure correct delivery of a message using headers of length $\lceil \log_2(f + 1) \rceil$ (see [1]).

Adler et al. [2] studied memoryless protocols for $m \times n$ meshes (grids), where m is any constant greater than 2. Although the feedback vertex sets of these graphs have size $\Omega(n)$, they prove that, for these graphs, packet headers of length $\Theta(\log\log n)$ are necessary and sufficient. In contrast, the $2 \times n$ mesh, which also has only linear size feedback vertex set, supports a memoryless protocol without headers that ensures correct delivery of a message. Fraigniaud and Gavoille [10] recently proved an $\Omega(n)$ lower bound on the packer header length for an $n \times n$ mesh and from this obtained a lower bound on the packet header length for any graph in terms of its tree-width. Adler and Fich [1] also give an algorithm to construct memoryless protocols for graphs that can be decomposed in a certain way, by combining protocols for the various components. It had remained an open problem to characterize the family of graphs for which packet headers of a particular size suffice.

Further elaboration of the model described here and descriptions of related models and results can be found in [1, 9]. These papers also describe related models, problems, and protocols more fully, including sending a sequence of messages from s to r and allowing information to be stored at intermediate nodes.

3. Basics for graphs.

In this section we review basic terminology and standard results used throughout this paper. We encourage readers unfamiliar with the basic graph-theoretic concepts to consult the introductory text of West [19], whose notation we usually follow. For convenience we collect the most relevant definitions here. In this paper, we consider simple nonempty graphs with a finite number of vertices.

The set of neighbors of a vertex v in a graph G, also called its *neighborhood*, is denoted by $N_G(v)$ or $N(v)$. The *closed neighborhood* of v is $N(v) \cup \{v\}$ and is denoted by $N_G[v]$ or $N[v]$. The number of edges incident to v is its *degree*, denoted by $d_G(v)$ or $d(v)$. Given a set $U \subseteq V(G)$, the subgraph *induced* by the vertex set U is the graph with vertex set U whose edge set consists of all the edges in G whose endpoints are contained in U. We denote this subgraph by $G[U]$ and the subgraph $G[V(G) - U]$ by $G - U$.

A *walk of length* k in a simple graph is a list of vertices $\langle v_0, v_1, \ldots, v_k \rangle$ such that $v_{i-1}v_i$ is an edge for $1 \leq i \leq k$. The *endpoints* of a walk are its first and last vertices. A walk is *closed* if its first vertex is the same as its last. A *path* on n vertices, denoted by P_n, is a simple graph G whose vertices can be ordered as $v_1, v_2, \ldots, v_n$ so that $E(G) = \{v_iv_{i+1} : 1 \leq i < n\}$. The vertices v_1 and v_n are the *endpoints* of the path. A *subpath* of P is a path contained in P; its vertices form a consecutive sublist of $v_1, v_2, \ldots, v_n$. A path with endpoints u and v that has m edges is a u, v*-path* of *length* m. A *cycle* on n vertices, denoted by C_n, is a simple graph G whose vertices can be ordered $v_1, v_2, \ldots, v_n$ so that $E(G) = \{v_iv_{i+1} : 1 \leq i < n\} \cup \{v_nv_1\}$. The $m \times n$ *grid* is the simple graph with vertex set $\{1, \ldots, m\} \times \{1, \ldots, n\}$ where two vertices are adjacent if they agree in one coordinate and differ by exactly one in the other coordinate. An $m \times n$ *mesh* is an $m \times n$ grid plus the vertices s and r, where s is adjacent to all grid vertices whose second entry is 1 and r is adjacent to all grid vertices whose second entry is n.

A graph is *connected* if, for every two vertices u and v, it contains a u, v-path. A maximal connected subgraph of a graph is a *component*. A *cut-set* of G is a set of vertices whose deletion increases the number of components of G. A *cut-vertex* or *cut-edge* of G is a vertex or edge whose deletion increases the number of components of G. A *block* of a graph is a maximal connected subgraph which has no cut-vertices. A connected graph G is k*-connected* if it has more than k vertices and every cut-set is of size at least k. The blocks with at least three vertices are the maximal 2-connected subgraphs.

When $X, Y \subseteq V(G)$, an *X, Y-path* is an x, y-path for some $x \in X$ and $y \in Y$; such a path is *strict* if it intersects X and Y only at its endpoint. We state a version of Menger's theorem, which we use frequently: if G is k-connected, then for any disjoint sets $X, Y \subseteq V(G)$ there are k internally disjoint strict X, Y-paths in G; moreover, if X (and Y) is of size k, then we may assume that every vertex in X (and Y) is the endpoint of exactly one path. We often apply this variation in a 3-connected graph to obtain three paths with a common endpoint.

4. Graph minors and *sr*-graphs.

In this section we collect a few definitions and results from the theory of graph minors and define the graph model that we study in this paper. For more detail on graph minors, we recommend Diestel's [7] text on graph theory.

Subdividing an edge uv is the process of replacing uv with a path u, w, v of length 2, where w is a new vertex. An *H-subdivision* is a graph G obtained from a graph H by a succession of subdivisions. In this case the vertices of H in G are the *branch vertices*. The edges of H are represented in G by internally disjoint paths joining the corresponding branch vertices.

Contracting an edge $uv \in E(G)$ (to a new vertex w) produces the simple graph $G \cdot uv$ by replacing u and v with a single vertex w adjacent to the neighbors of u and v in $G - uv$. We also extend the notation so that we may contract a connected subgraph (to w), meaning that we contract all edges within it, leaving only one vertex (called w). A *minor* of G is a graph that can be obtained from a subgraph of G by edge contractions. If H is a minor of G, then let a *model of H* be a subgraph of G that represents H as follows: each vertex $v \in V(H)$ is represented by a set of vertices that induces a connected graph in G called the *branch set* for v, and each edge in H is represented in the model by an edge in G joining the corresponding branch sets. The vertex set of the model is partitioned into the branch sets. Each edge in the model either represents an edge in H or has both of its endpoints in a single branch set. The model can be contracted to H by contracting every edge that doesn't represent an edge in H. If H is a minor of G and G is connected, then G has a spanning model of H.

A *rooted graph* is a graph with a list of distinguished vertices called its *roots*. Let G be a rooted graph with roots $x_1, \ldots, x_k$, and let H be a rooted graph with roots $y_1, \ldots, y_k$. H is a *rooted minor* of G if G contains a model of H such that for all i, x_i is contained in the branch set for y_i; such a model is a *rooted model of H*. The graph model that we study in this paper is the special case when we have exactly two roots s and r.

DEFINITION 4.1. *An sr*-rooting *of a graph G is a designation of one vertex of G to be the sender s and another vertex to be the receiver r. An sr-rooted connected graph will be called simply an sr*-graph. *An sr-graph H is an sr*-minor *of another sr-graph G if H is a rooted minor of G, that is, G contains a model of H and the branch set in G for the vertex s (respectively, r) of H contains the vertex s (respectively, r) of G.*

The following basic result of Adler and Fich [1] establishes a close relation between reliable single message transmission and rooted minors.

PROPOSITION 4.2. *If an sr-graph G allows reliable single message transmission with headers of size d, and H is an sr-minor of G, then there is a modification of the protocol used for G that allows reliable single message transmission with headers of size d in H.*

This proposition will allow us to prove that there is a forbidden minor characterization for *sr*-graphs in which reliable single message transmission with headers of size at most d is possible.

DEFINITION 4.3. *A family of sr-graphs $\mathcal{F}$ is an* antichain (in the *sr*-minor-order) *if no graph in $\mathcal{F}$ is an sr-minor of another graph in $\mathcal{F}$. A* forbidden minor characterization *of a family of sr-graphs $\mathcal{G}$ is an antichain $\mathcal{F}$ such that an sr-graph is in $\mathcal{G}$ if and only if it does not have an sr-minor in $\mathcal{F}$.*

It is a simple observation that if a family of *sr*-graphs has a forbidden minor characterization, then this characterization is unique. The next result follows directly from the seminal work of Robertson and Seymour on graph minors. If Ω is chosen to be an appropriate well-quasi-order on the k roots in the proof of Proposition 10.5. of [15], then this result follows along the same lines [18].

THEOREM 4.4. *If $\mathcal{F}$ is a family of rooted graphs with k roots, none of which is a rooted minor of another, then $\mathcal{F}$ is finite.*

We now easily obtain the following result, which is a crucial ingredient in the recognition algorithm for *sr*-graphs allowing reliable single message transmission with headers of size d.

THEOREM 4.5. *For every d, the set of sr-graphs allowing reliable single message transmission with headers of size d has a unique finite forbidden minor characterization, $\mathcal{F}_d$.*

Proof. By the preceding remarks it suffices to exhibit a forbidden minor characterization $\mathcal{F}_d$. Let $\mathcal{F}_d$ be the family of *sr*-graphs F such that F does not permit reliable single message transmission with headers of size d, but every proper *sr*-minor of F does. It is immediate that $\mathcal{F}_d$ is an antichain. If G is an *sr*-graph that does not allow reliable single message transmission with headers of size d, then it follows from the definition of $\mathcal{F}_d$ that G contains a member of $\mathcal{F}_d$ as an *sr*-minor. The converse of this statement follows by Proposition 4.2, so that $\mathcal{F}_d$ is a forbidden minor characterization. □

Our main result is that $\mathcal{F}_0$ is the set of *sr*-graphs in Figure 3 in section 8.

5. Cut-vertices, and *sr*-splits.

Next we start developing tools to study the structure of *sr*-graphs in $\mathcal{F}_d$ (for any fixed d). By definition, s and r must be in the same component of any graph in $\mathcal{F}_d$; otherwise there is no possible path from s to r. We show next that such a graph is in fact 2-connected. In a given *sr*-graph, let B^{uv} be the block containing u and v. Let B^{uv} be *sr*-rooted such that u plays the role of s and v plays the role of r.

LEMMA 5.1. *Let $v_1, v_2, \ldots, v_k$ be the unique sequence of cut-vertices encountered on every sr-path of an sr-graph G. Reliable single message transmission (with headers of size d) from s to r is possible in G if and only if it is possible in $B^{v_iv_{i+1}}$ for $0 \leq i \leq k$, where $v_0 = s$ and $v_{k+1} = r$.*

Proof. The graph $B^{v_iv_{i+1}}$ is an *sr*-minor of G with s in the branch set corresponding to v_i and r in the branch set corresponding to v_{i+1}. By Proposition 4.2, it follows that if there is a protocol (using headers of size d) that ensures correct delivery of a message from s to r in the graph G, then there is a protocol using headers of size d that ensures correct delivery of a message from v_i to v_{i+1} in the graph $B^{v_iv_{i+1}}$.

Conversely, suppose there exists a protocol A_i using headers of size d that ensures correct delivery of a message from v_i to v_{i+1} in the graph $B^{v_iv_{i+1}}$ using headers of size d. We may assume that, in A_i, v_{i+1} never sends packets to its neighbors in $B^{v_iv_{i+1}}$, and v_i never receives packets from its neighbors in $B^{v_iv_{i+1}}$. A protocol for correct delivery of a message from s to r in G can be obtained by combining $A_0, \ldots, A_k$ as follows. In the combined protocol, a cut-vertex v_i takes each incoming message from $B^{v_{i-1}v_i}$ and forwards it into $B^{v_iv_{i+1}}$ just as A_i would forward a message originating at v_i. All other vertices will use the protocol from the unique block to which they belong.

There is a constant c_i for each $B^{v_i v_{i+1}}$ that bounds the number of copies of a message originating at v_i that A_i will generate, no matter which edges of $B^{v_i v_{i+1}}$ are operational. Therefore the new protocol will generate at most $c_0 \cdots c_k$ copies of a message originating at s. In this protocol, a message sent by s will be received by r, no matter what links are operational, as long as there is always a path of operational links between s and r. ∎

Lemma 5.1 shows that every graph in $\mathcal{F}_d$ is 2-connected for any d. If G is a 2-connected *sr*-graph, then for every minor of G, the 2-connectedness will allow us to modify its model to obtain an *sr*-minor. We make this more precise now. As remarked before, if G contains H as an *sr*-minor, then G must contain H as a minor. In hunting for minors, we either delete or contract edges. The next definition introduces an inverse operation to contracting an edge sr.

DEFINITION 5.2. *An* sr-split *of an unrooted connected graph G is the sr-graph obtained by replacing some vertex v of degree at least* 4 *by two new adjacent vertices s and r with the property that $N(s) \cap N(r) = \emptyset$, $N(s) \cup N(r) = N(v) \cup \{s, r\}$, and $|N(s) - r|, |N(r) - s| \geq 2$.*

Clearly, if G is an *sr*-split of H, then G has H as a minor. The following lemma establishes a partial converse.

LEMMA 5.3. *Let G be a* 2*-connected graph. The following are equivalent for a graph H:*

(i) *G has H as a minor;*

(ii) every *sr-rooting of G has an sr-rooting or an sr-split of H as an sr-minor.*

Proof. Given (ii), fix an *sr*-rooting of G. If an *sr*-split of H is an *sr*-minor of G, then G contains a rooted model of an *sr*-split of H. We contract the edge sr of the split to obtain H and merge the branch sets for s and r together with the path corresponding to the edge sr to obtain a model of H in G. If an *sr*-rooting of H is an *sr*-minor of G, then by definition H is a minor of G.

To see that (i) implies (ii), let G be *sr*-rooted. For every $v \in V(H)$, let S_v be the branch set that corresponds to v in a spanning model of H of G. We wish to modify the branch sets so that they are branch sets of a rooted model of an *sr*-rooting or an *sr*-split of H in G. Now, if $s \in S_u$ and $r \in S_v$, for $u \neq v$, then we label u with s and v with r to obtain the desired *sr*-rooting of H with a rooted model of G. Hence we may assume that $s, r \in S_v$ for some v. By 2-connectedness, there are two disjoint paths P_s, P_r from $\{s, r\}$ to $V(G) - S_v$. Let T be a spanning tree of $G[S_v]$ that contains the portion of these paths within $G[S_v]$. Let e be an edge on the shortest path in T from P_s to P_r, and let T_s and T_r be the components of $T - e$ containing s and r, respectively. Thus $V(T_s)$ and $V(T_r)$ each have a neighbor in a branch set other than S_v. Let m_s be the number of vertices in $N_H(v)$ whose branch sets in G contain a neighbor of $V(T_s)$ but not a neighbor of $V(T_r)$. Define m_r similarly by interchanging s and r.

First suppose that $m_s \geq 2$ and $m_r \geq 2$. We obtain a rooted model in G of an *sr*-split of H by replacing S_v with branch sets $V(T_s)$ for s and $V(T_r)$ for r. Therefore, by symmetry we may assume that $m_s \leq 1$. If $m_s = 1$, then let u be the vertex that is counted by m_s. If $m_s = 0$, then let u be any vertex, other than v, whose branch set contains a neighbor of $V(T_s)$. By the construction, $S_v - V(T_s)$ and $S_u \cup V(T_s)$ induce connected graphs, and e is an edge joining these sets. Move $V(T_s)$ from S_v to S_u (see Figure 1). The resulting branch set for v is adjacent to all branch sets for vertices in $N_H(v)$. These new branch sets define a spanning model of H such that s and r are in different branch sets, which reduces this case to the first case. ∎

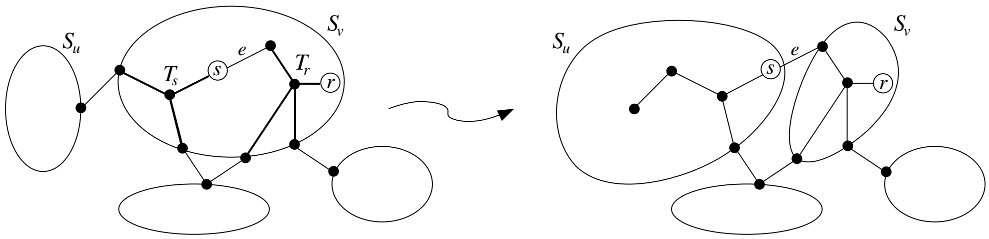


FIG. 1. *Branch sets for vertices in* $N[v]$ *before and after the move, where* $m_s = 1$.

The conclusion of Lemma 5.3 need not hold when G is not 2-connected. For example, let G be a triangle plus s and r adjacent to the same vertex of the triangle, and let H be a triangle. Although H is a minor of G, an *sr*-rooting of a triangle is not an *sr*-minor of G. Also, H has no *sr*-split, so H has no *sr*-split that yields G.

6. Algorithms for reliable single message transmission. In this section we discuss algorithms for reliable single message transmission in *sr*-graphs. It is easy to quickly decompose a graph into its blocks [19, p. 157]. We proved in Lemma 5.1 that to determine if transmission from s to r is possible with headers of size d, it is sufficient to study each block that is encountered on an *sr*-path. This enables us to restrict our attention to 2-connected graphs. We will prove that there is a linear-time algorithm to test for the feasibility of reliable single message transmission with headers of size d in 2-connected graphs. The proof uses the notion of tree-width, which is a measure of how "tree-like" a graph is. For a formal definition of tree-width, see Definition 14.1.

Many hard problems can be solved efficiently for graphs of bounded tree-width. Connected graphs of tree-width 1 are trees. In a tree, every pair of nodes is joined by a unique path, so no headers are needed for reliable transmission. In section 8, we will see that headers also are not needed for graphs of tree-width 2. The result of Adler et al. [2] mentioned in section 2 shows that already for graphs of tree-width 3 there is no constant upper bound on the length of the headers needed since a $3 \times n$ mesh has treewidth at most 3. On the other hand, reliable single message transmission with bounded header-size requires small tree-width.

THEOREM 6.1. *For every* d *there is a constant* c_d *such that if* G *is a* 2*-connected sr-graph with tree-width greater than* c_d*, then reliable single message transmission from* s *to* r *with headers of length* d *is not possible in* G.

Proof. Let $G_{3,n}$ denote the $3 \times n$ mesh. By [2] we can find $n \geq 9$ such that in $G_{3,n-2}$ we do not have reliable single message transmission with headers of size d. Let $c_d = 400^{n^5}$. By a result of Robertson, Seymour, and Thomas [17] (improving on results from [14]), graphs with tree-width greater than c_d have an $n \times n$ grid as a minor. Since G is a 2-connected graph with an $n \times n$ grid as a minor, Lemma 5.3 implies that G has an *sr*-rooting or an *sr*-split of the $n \times n$ grid as an *sr*-minor. Since every *sr*-rooting and *sr*-split of an $n \times n$ grid for $n \geq 9$ contains $G_{3,n-2}$ as an *sr*-minor, we cannot have reliable single message transmission with headers of length d in G. □

We will now use this to show that there is a linear time algorithm for recognizing graphs that allow reliable single message transmission with headers of fixed size.

THEOREM 6.2. *For fixed* d*, there is an* $O(n)$ *algorithm that determines whether reliable single message transmission with headers of size* d *is possible in* 2*-connected sr-graphs on* n *vertices.*

Proof. For a given *sr*-graph G, it suffices to check whether G contains one of the *sr*-graphs from $\mathcal{F}_d$ as an *sr*-minor. We apply an $O(n)$-time algorithm due to Bodlaender [6] that either determines that G has tree-width at least c_d+1 or produces a tree-decomposition of width at most c_d. In the former case, reliable single message transmission with headers of size d is not possible, by Theorem 6.1. If G has a tree-decomposition of width at most c_d, then there is an $O(c_d n)$-time algorithm by Arnborg, Lagergren, and Seese [4] that checks if G has a specified *sr*-graph as an *sr*-minor. We apply this algorithm at most $|\mathcal{F}_d|$ times to check whether any *sr*-graph in $\mathcal{F}_d$ is an *sr*-minor of G. Since c_d and $|\mathcal{F}_d|$ are constant for fixed d, altogether this takes time that is linear in n. □

Since the algorithm in Theorem 6.2 uses the forbidden minor characterization $\mathcal{F}_d$ which we do not know for all values of d, it is unfortunately not implementable in general. However, in Theorem 8.1 we determine $\mathcal{F}_0$. Furthermore, in Corollary 9.6 we prove that any graph that allows reliable headerless single message transmission has tree-width at most 3. Using Bodlaender's algorithm [6] (or an algorithm due to Matoušek and Thomas [11]) we can obtain a tree-decomposition of tree-width at most 3, which can be quickly checked (using [4]) to determine whether the network has any *sr*-minors from $\mathcal{F}_0$. Thus in the headerless case we can find the guaranteed algorithm.

7. Headerless transmission and fragments.

In the remaining sections, we will focus on the question of when reliable single message transmission is possible in an *sr*-graph G using no headers. By Theorem 4.5, there is a unique antichain $\mathcal{F}_0$ such that headerless reliable single message transmission in G is possible if and only if G does not have any *sr*-graph from $\mathcal{F}_0$ as an *sr*-minor.

DEFINITION 7.1. *An sr-graph in* $\mathcal{F}_0$ *is an* obstruction*, and* $\mathcal{F}_0$ *is called the* obstruction set*. An obstruction is* minor-minimal *in the sense that every proper sr-minor of it allows headerless reliable single message transmission.*

Our next theorem characterizes *sr*-graphs that allow reliable headerless single message transmission in terms of a structural property.

DEFINITION 7.2. *A* closed walk *in a graph is a list of vertices* $\langle y_0, y_1, y_2, \ldots, y_k\rangle$ *(not necessarily distinct) such that consecutive vertices are adjacent and* $y_0 = y_k$*. The indices here and later are always taken modulo* k*. A* fragment circuit π *of an sr-graph* G *is a closed walk* $\langle y_0, y_1, y_2, \ldots, y_k\rangle$ *such that for all* $1 \le i \le k$*,* $y_{i-1}y_iy_{i+1}$ *is a path in* G *and is a subpath of some sr-path* P_i *in which the three vertices appear in the given order when* P_i *is traversed from* s *to* r*.*

THEOREM 7.3. *There is a protocol for without using headers reliable headerless single message transmission in an sr-graph* G *if and only if* G *has no fragment circuit.*

Proof. Suppose that G contains a fragment circuit $\langle y_0, \ldots, y_k\rangle$. For $1 \le i \le k$, let P_i be an *sr*-path containing the subpath $y_{i-1}y_iy_{i+1}$ in that order. Since each P_i might be the only operational *sr*-path, any message that travels along $y_{i-1}y_i$ for some i must be forwarded along y_iy_{i+1}. Furthermore, in the case that no edge of G fails, a message must be sent along P_1 and thus travels along y_0y_1. This message must then be forwarded along $y_1y_2, y_2y_3, \ldots$. Since the walk is closed, this forces an infinite amount of packet traffic when no edge of G fails.

Conversely, suppose there are no fragment circuits. Consider the following generic protocol. When node v receives a message from its neighbor u, it sends a copy of the message to its neighbor w if and only if uvw appears in that order along an *sr*-path. If P is an operational *sr*-path, then this will transmit the message from s to r along P.

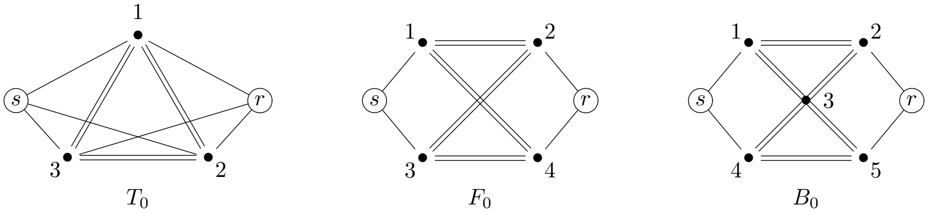


FIG. 2. *Three obstructions.*

To prove that there is only a finite amount of packet traffic generated from one message, we suppose otherwise. Since G is finite, there are finitely many sr-paths in G. Hence there is a bound on the number of packets that are forwarded each time a message is received at a node. Hence there must be packets that follow walks of arbitrary length starting from s. However, a sufficiently long walk must repeat some edge in the same direction. Say such a walk begins with an initial segment α from s, followed by the vertices $v_0, v_1, \ldots, v_{k-1}, v_k, v_{k+1}$, where $v_0 = v_k$ and $v_1 = v_{k+1}$. By definition of the protocol, v_{i-1}, v_i, v_{i+1} is a subpath of an sr-path for $i = 1, \ldots, k$. This contradicts the assumption that G contains no fragment circuit. □

We note that obtaining the protocol given in the proof may be computationally expensive since there may be exponentially many sr-paths. Since we wish to determine the obstruction set, we will investigate graphs with fragment circuits. We use the following terminology in studying these graphs.

DEFINITION 7.4. *Let $\langle y_0, y_1, \ldots, y_k\rangle$ be a fragment circuit in an sr-graph G. We call the path $y_{i-1}y_iy_{i+1}$ a* fragment *and designate it π_i. Let $P_1, P_2, \ldots, P_k$ be fixed sr-paths associated with the fragment circuit (where π_i is a subpath of P_i). We call this collection $\langle P_1, P_2, \ldots, P_k\rangle$ a* fragment structure *and refer to each of its members as a* fragment path. *With π designating a fragment circuit, the graph G_π with vertex set $\{y_i : 0 \leq i \leq k-1\}$ and edge set $\{y_{i-1}y_i : 1 \leq i \leq k\}$ is the* fragment subgraph *of G for π. Note that G may have more than one fragment subgraph.*

If in a fragment circuit the vertices $y_{i-1}, \ldots, y_{j+1}$ (for $i < j$) form a subpath of a single sr-path P in which the vertices appear in the given order as P is traversed from s to r, then we will sometimes specify all of the fragments $\pi_i, \ldots, \pi_j$ at once by referring to $y_{i-1} \ldots y_{j+1}$ as an extended fragment *with fragment path P.*

The following examples illustrate Definition 7.4.

Example 7.5. Figure 2 shows three sr-graphs T_0, F_0, and B_0 with fragment circuits highlighted by drawing double edges. Note that T_0 is the graph K_5 minus the edge sr and F_0 is the graph $K_{3,3}$ minus the edge sr.

In T_0, the fragment circuit 1231 consists of the fragments 123, 231, 312. Here P_1 is $s123r$, P_2 is $s231r$, and P_3 is $s312r$. We name this graph T_0 because the fragment subgraph is a triangle.

In F_0, the fragment circuit 12341 consists of the extended fragments 1234 and 3412, with the fragment subgraph forming a 4-cycle. In B_0, the fragment circuit 1234531 consists of the extended fragments 12345 and 45312, with the fragment subgraph forming a bowtie. The notations F_0 and B_0 reflect the initials of "4-cycle" and "bowtie."

By Theorem 7.3, the presence of these fragment circuits implies that reliable headerless single message transmission is not possible from s to r in these graphs. In fact, these graphs are obstructions: contracting or deleting any edge results in a graph admitting reliable single message transmission from the resulting s to r. For example, in T_0, since a fragment must use three vertices in $G - \{s, r\}$, contracting any edge or deleting an edge from the triangle results in an *sr*-graph that cannot have a fragment cycle. Also, if we delete the edge from s to 1, then 1 cannot be the initial point on a fragment, so 1 cannot be on a fragment circuit. The same argument applies for removing any of the edges incident with s or r. Similar case analysis for F_0 and B_0 shows that these are also obstructions.

The next remarks collect a few simple but useful observations about fragment subgraphs.

Remark 7.6. Every vertex in a fragment subgraph is the midpoint of at least one fragment and is an endpoint of at least two fragments. Neither s nor r can be the midpoint of a fragment, and so a fragment subgraph of an *sr*-graph G is a subgraph of $G - \{s, r\}$. Furthermore, a fragment subgraph is connected and has minimum degree at least 2. Clearly, the edge *sr* cannot be in a fragment path; hence if G is an obstruction, then *sr* is not an edge in G.

Remark 7.7. Definition 7.4 is symmetric if the labels s and r are switched. Let G' be the *sr*-graph obtained from G by switching s and r. Then $\langle y_0, \ldots, y_k\rangle$ is a fragment circuit in G if and only if $\langle y_k, \ldots, y_0\rangle$ is a fragment circuit in G'. Thus a fragment subgraph of G is also a fragment subgraph of G'. It follows that G allows reliable headerless single message transmission if and only if G' allows reliable headerless single message transmission.

8. The main result and its consequences.

Figure 3 contains 10 *sr*-graphs in which reliable headerless single message transmission is not possible due to the fragment circuits indicated. (These fragment circuits are not unique; for example, F_1 has a fragment circuit of length 3.) Let $\mathcal{F}_0'$ denote the set of these *sr*-graphs. Our goal in the remaining sections is to show that the obstruction set $\mathcal{F}_0$ is exactly $\mathcal{F}_0'$.

THEOREM 8.1. *$\mathcal{F}_0'$ is the unique antichain with the property that an sr-graph permits reliable headerless transmission of a single message if and only if it contains no sr-minor from $\mathcal{F}_0'$.*

We begin the proof of Theorem 8.1 by noting that it is straightforward but somewhat tedious to show that no element of $\mathcal{F}_0'$ is an *sr*-minor of any other, i.e., that $\mathcal{F}_0'$ is an antichain. (For full details, see [12].) It follows from Proposition 4.2 that no *sr*-graph with an *sr*-minor from $\mathcal{F}_0'$ allows headerless single message transmission. To show that $\mathcal{F}_0'$ is the desired forbidden minor characterization (and therefore unique) it remains to prove that every *sr*-graph which does not allow headerless single message transmission contains a minor in $\mathcal{F}_0'$. However, every such *sr*-graph contains an obstruction as an *sr*-minor, so that in fact it suffices to prove the following.

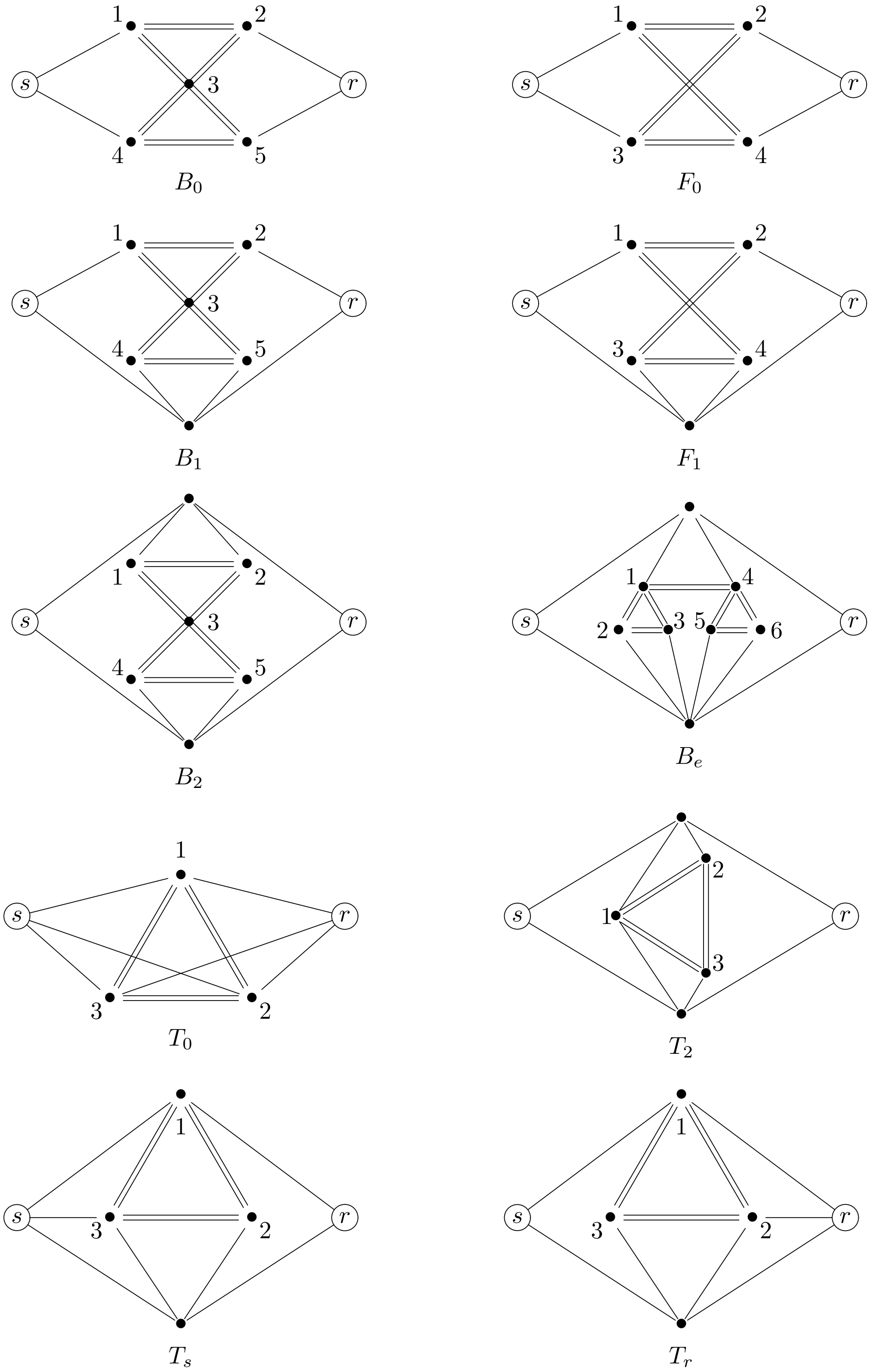
THEOREM 8.2. *Every obstruction $G \in \mathcal{F}_0$ has some $H \in \mathcal{F}_0'$ as an sr-minor.*

We devote the remaining sections to the proof of this theorem. In the rest of this section, we present some corollaries that follow from Theorem 8.1.

COROLLARY 8.3. *If G is a graph of tree-width at most 2, then reliable headerless single message transmission is possible in every sr-rooting of G.*

Proof. It is well known that a graph has with tree-width at most 2 if and only if it does not have K_4 as a minor (see [7, p. 263]). However, each of the graphs in $\mathcal{F}_0'$ contains a subdivision of K_4, so no obstruction can be an *sr*-minor of an *sr*-graph of tree-width at most 2. □

We now apply Theorem 8.1 to a related problem that we call the *broadcasting* problem. In this model, we have only one distinguished vertex s, the source, from which we wish to broadcast to all other vertices in the graph. We assume that this single-root graph is connected and we call it an s-graph. The main assumptions of



B_0

F_0

B_1

F_1

B_2

B_e

T_0

T_2

T_s

T_r

FIG. 3. *The family* $\mathcal{F}_0'$.

the model remain the same as described in section 1 for the reliable single message transmission problem. Edges can fail, and this may lead to the graph being disconnected, but we require only that every vertex that is reachable from s through some path receives the message. A successful protocol to broadcast in G ensures that every vertex of G that is in the same component as s receives the message, and none of them gets flooded by infinitely many copies.

The following proposition relates the broadcasting model to the reliable single message transmission model. Let $G \vee r$ be the sr-graph obtained from an s-graph G by adding to it a new vertex r and making it adjacent to every vertex in G except s.

PROPOSITION 8.4. *Any protocol that can transmit reliably from s to r in $G \vee r$ can be modified to obtain a protocol that can broadcast successfully in G, and conversely.*

Proof. For any vertex $v \neq s$ in G, it may be that the only operational path from s to r in $G \vee r$ uses the edge vr. Thus a protocol that can transmit reliably from s to r in $G \vee r$ must be able to transmit to each neighbor of r that can be reached from s. That is, it is a protocol to broadcast in G.

Conversely, consider the following protocol to transmit from s to r in $G \vee r$ that makes use of a successful broadcast protocol in G. A vertex in G always passes the message to r when it receives it, along with whatever else the protocol tells it to do. Thus the message will reach r if any sr-path remains. Since the broadcast protocol ensures that each vertex receives only a finite number of copies of the message, r also receives only finitely many copies of the message in the transmission protocol. □

We say that an s-graph H is an *s-minor* of another s-graph G if H is a rooted minor of G (that is, G contains a model of H and the branch set of G for the vertex s of H contains the vertex s of G). We can now use Theorem 8.1 to obtain a characterization of graphs that do not permit headerless broadcasting.

COROLLARY 8.5. *There is a protocol to broadcast without headers in an s-graph G if and only if G has no s-minor from $\{F_0 - r, T_0 - r, B_0 - r\}$.*

Proof. By Proposition 8.4 it suffices to characterize s-graphs G such that headerless reliable single message transmission is possible in $G \vee r$. If G contains one of the three given s-minors, then by Theorem 8.1 headerless reliable single message transmission is not possible in $G \vee r$. Conversely, if $G \vee r$ does not allow headerless reliable single message transmission, it must contain an sr-minor $H \in \mathcal{F}_0'$. Thus G contains $H - r$ as an s-minor and it is easy to check that contain $T_0 - r, F_0 - r$, and $B_0 - r$ as s-minors, respectively. □

Remark 8.6. Observe that $T_0 - r$ is the graph K_4 with a vertex labeled s. As mentioned in the proof of Corollary 8.3, a graph has tree-width at most 2 precisely when it does not have K_4 as a minor. Thus headerless broadcast in an s-graph G is only possible if G is in a restricted class of s-graphs of tree-width at most 2.

9. Tree-width of $G + sr$.

Often we will consider the graphs $G - \{s, r\}$ and $G + sr$ for an obstruction G. By Remark 7.6 we know something about their structure. For convenience, we introduce the following notation.

DEFINITION 9.1. *Suppose that G is an sr-graph such that $sr \notin E(G)$. We let G^+ denote the sr-graph $G + sr$ obtained by adding the edge sr, and we let G^- denote the graph $G - \{s, r\}$ obtained by deleting the vertices s and r.*

In this section, we establish that if G is an obstruction other than T_0, then G^+ has tree-width at most 3. This restricts the structure of G and will be used in the last three sections. We start with two simple observations.

LEMMA 9.2. *An sr-rooting of the 3-cube Q_3 in which s and r are in opposite partite sets has F_0 or B_0^+ as a proper sr-minor.*

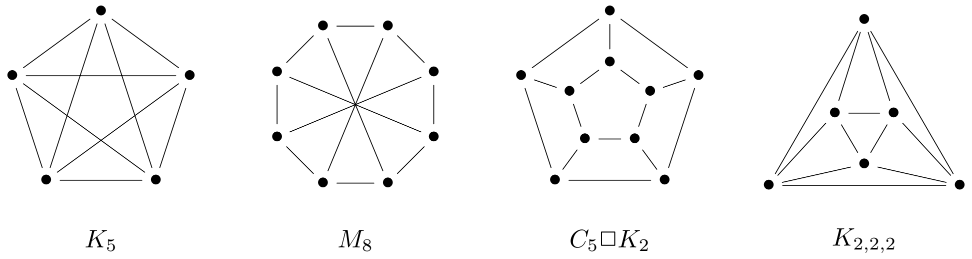


FIG. 4. *Forbidden minors for tree-width ≤ 3.*

Proof. If s and r are at distance 3 in the cube (that is, they are antipodal), then the remaining vertices form a 6-cycle. Contracting three consecutive vertices on the 6-cycle to a single vertex (and then deleting two appropriate edges) yields F_0.

If s and r are adjacent vertices in Q_3, then we obtain B_0^+ as a proper sr-minor by contracting the edge antipodal to sr. □

COROLLARY 9.3. *If G is an obstruction, then G^+ does not have as an sr-minor any sr-rooting of the 3-cube Q_3 in which s and r are in opposite partite sets.*

Proof. If there were such a Q_3, then Lemma 9.2 would contradict the minor-minimality of G. □

We will now apply a theorem of Arnborg, Corneil, and Proskurowski [3], which makes an important connection between minors and tree-width.

THEOREM 9.4. *A graph has tree-width at least 4 if and only if it has as a minor at least one of the complete graph K_5, the Möbius ladder M_8 of order 8, the pentagonal prism $C_5 \square K_2$, or the octahedron $K_{2,2,2}$. (See Figure 4.)*

THEOREM 9.5. *If G is a 2-connected sr-graph that has tree-width at least 4, then G has T_0, F_0, B_0^+, or T_r^+ as a proper sr-minor.*

Proof. By Lemma 5.3 it suffices to check that every sr-rooting and sr-split of the graphs in Theorem 9.4 has at least one of $\{T_0, F_0, B_0^+, T_r^+\}$ as a proper sr-minor. □

COROLLARY 9.6. *If G is 2-connected and has tree-width at least 4, then reliable headerless single message transmission is impossible for any sr-rooting of G.*

Proof. The proof is immediate from Theorem 9.5 and the fact that the graphs in $\mathcal{F}_0'$ do not allow reliable headerless single message transmission. □

THEOREM 9.7. *If G is an obstruction such that G^+ has tree-width at least 4, then $G = T_0$.*

Proof. By the remark after Lemma 5.1, G^+ is 2-connected. By Theorem 9.5 it must have T_0, F_0, B_0^+, or T_r^+ as a proper sr-minor. Hence we may assume that $G^+ \in \{T_0^+, F_0^+\}$, since otherwise we contradict the minor-minimality of G. Since F_0^+ is an sr-rooting of $K_{3,3}$ and it is well known that the tree-width of $K_{3,3}$ is 3, the only remaining case is $G = T_0$. □

10. 2-cuts in obstructions. We have seen in previous sections that if G is an obstruction other than T_0, then G^+ is 2-connected and has tree-width 3. Graphs that are 3-connected and have tree-width 3 have many nice properties, as we will see in section 14. For example, certain NP-hard problems become tractable when restricted to graphs with bounded tree-width [4]. In this section, we study 2-cuts in obstructions with the aim of proving that obstructions are almost 3-connected.

Throughout this section, G is an obstruction, $X = \{x_1, x_2\}$ is a cut-set in G, and $\pi = \langle y_0, y_1, \ldots, y_k \rangle$ is a fixed fragment circuit in G. Furthermore, let C_s and C_r

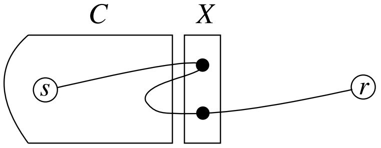


FIG. 5. *The path P in Lemma* 10.3 *if $P - V(C)$ is not a path.*

denote the components of $G - X$ that contain s and r, respectively (possibly $C_s = C_r$, or these could be empty if $s, r \in X$).

LEMMA 10.1. *Let G be an obstruction, and let X be a 2-cut in G. If $y_i \in X$, then $y_{i-1} \in V(C_s)$ or $y_{i+1} \in V(C_r)$.*

Proof. Let P_i be an sr-path containing $y_{i-1}y_iy_{i+1}$ in this order. Suppose that $y_i = x_1$, so we can write P_i as $sPx_1P'r$. If sP is contained in one component of $G - X$, then $y_{i-1} \in V(C_s)$. If sP is not contained in one component of $G - X$, then x_2 appears on sP. Thus $P'r$ is contained in one component of $G - X$, so that $y_{i+1} \in V(C_r)$. □

LEMMA 10.2. *Let G be an obstruction. If X is a 2-cut in G and $C_s \neq C_r$, then G_π is completely contained in one of $C_s \cup X, C_r \cup X$, or some other component C of $G - X$.*

Proof. If $V(G_\pi)$ does not intersect X, then G_π is entirely contained in one component of $G - X$. So consider $y_i \in X$. By Lemma 10.1 and symmetry (Remark 7.7), we may assume that $y_{i+1} \in C_r$. To obtain a contradiction, suppose that $V(G_\pi) \nsubseteq C_r \cup X$. Consider the first j cyclically after i such that $y_{j+1} \notin C_r \cup X$. Now $y_j \in X$ and $y_{j-1} \in C_r \cup X$, but this contradicts Lemma 10.1. □

LEMMA 10.3. *Let G be an obstruction with fragment subgraph G_π, and let X be a 2-cut in G. If C is a component of $G - X$ that does not intersect G_π, then $V(C) = \{s\}$ or $V(C) = \{r\}$ or $\{s, r\} \subseteq V(C)$.*

Proof. If $V(C)$ contains neither s nor r, then we could contract all of C into x_1 or x_2 without affecting G_π (the sr-paths P_i would merely be shortened), contradicting the minor-minimality of G. Suppose that $V(C)$ contains s but not r. Let P be an sr-path. A component of $P - V(C)$ must intersect X, so if $P - V(C)$ is not a path (see Figure 5), then it must consist of an isolated vertex in X plus a path from X to r. If P_i is a fragment path, then, since C does not intersect G_π, $y_{i-1}y_iy_{i+1}$ must be contained in $P_i - V(C)$. So $y_{i-1}y_iy_{i+1}$ is contained in a subpath P'_i from X to r. Thus we can contract all of C to s, replacing each fragment path P_i with sP'_i. Since this preserves a fragment circuit, minor-minimality of G implies that $V(C) = \{s\}$. By Remark 7.7, if $V(C)$ contains r but not s, then $V(C) = \{r\}$. □

LEMMA 10.4. *Let G be an obstruction. If X is a 2-cut in G, then $s, r \notin X$.*

Proof. Suppose that $s = x_1 \in X$. If $x_2 = r$, then G^- would be disconnected and only one of its components could intersect G_π, a contradiction to Lemma 10.3. So we may assume that $C_r \neq \emptyset$. Now $G - X$ contains another component C, that must intersect G_π by Lemma 10.3. By Lemma 10.2, C completely contains G_π, so $C_r = \{r\}$ by Lemma 10.3. Since G is 2-connected, $N(r) \subset X$ forces s and r to be adjacent in G, contradicting Remark 7.6. Hence $s \notin X$. Similarly, $r \notin X$. □

LEMMA 10.5. *Let G be an obstruction. If X is a 2-cut and $C_s \neq C_r$, then $C_s = \{s\}$ or $C_r = \{r\}$.*

Proof. By Lemma 10.4, both C_s and C_r are nonempty. By Lemma 10.2, one of them does does not meet G_π, say C_s. By Lemma 10.3, $C_s = \{s\}$. □

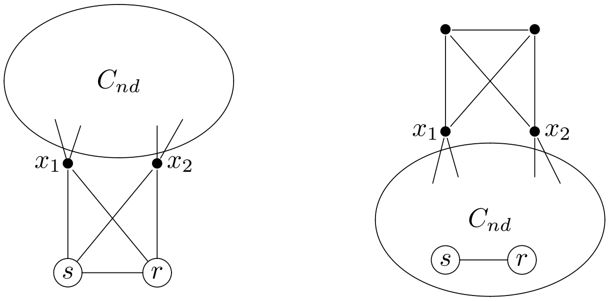


FIG. 6. *An sr-diamond-cut and a diamond-cut.*

LEMMA 10.6. *If G is an obstruction, and $\{x_1, x_2\}$ is a 2-cut, then $x_1x_2 \notin E(G)$.*

Proof. To obtain a contradiction, suppose that $x_1x_2 \in E(G)$. First we consider the case where $C_s \neq C_r$. By Lemma 10.5, either $C_s = \{s\}$ or $C_r = \{r\}$. By symmetry, we may assume that $C_s = \{s\}$. Now $N(s) = \{x_1, x_2\}$, since G is 2-connected. If $x_1x_2 \in E(G_\pi)$, then $\{x_1, x_2\} = \{y_i, y_{i+1}\}$ for some i, but this contradicts Lemma 10.1. So x_1x_2 is not in any fragment. If x_1x_2 is in a fragment path P_i, then P_i must begin sx_1x_2 or sx_2x_1. We can omit the second vertex of P_i and obtain a fragment path for the same fragment. Thus x_1x_2 can be deleted while maintaining a fragment circuit, contradicting minor-minimality.

Now suppose that $C_s = C_r$. Since G is 2-connected we may let P be an x_1, x_2-path through a component of $G - X$ other than C_s. Suppose that P_i is a fragment path that uses the edge x_1x_2 (in either direction). The only component of $G - \{x_1, x_2\}$ that P_i can intersect is C_s. Hence we can replace the edge between x_1 and x_2 by the path P (in the direction needed) to obtain a path P_i' in $G - x_1x_2$. We define a fragment or extended fragment π_i' in each path P_i' as follows: if a fragment $y_{i-1}y_iy_{i+1}$ contains the edge x_1x_2, then let π_i' be the subpath of P_i' created from the fragment by replacing x_1x_2 with P; otherwise let π_i' be $y_{i-1}y_iy_{i+1}$. Similarly, on the fragment circuit replace each occurance of x_1x_2 and x_2x_1 by P (in the direction needed); in the resulting sequence, any three consecutive vertices appear in order on some π_i'. This creates a new fragment structure and fragment circuit. Thus x_1x_2 can be deleted in this case as well, contradicting the minor-minimality of G. □

11. Diamond-cuts.

We will now consider 2-cuts in G^+, where G is an obstruction. Our aim will be to show that each 2-cut looks like one of the pictures in Figure 6.

Let X be a 2-cut in G^+. Since $G = G^+ - sr$, X is also a 2-cut in G. By Lemma 10.4, X contains neither s nor r. Also, s and r are together in some component C_{sr} of $G^+ - X$, where $C_{sr} = C_s \cup C_r + sr$. The lemmas of the previous section apply to X as a cut-set of G, but some also apply directly to G^+. In particular, let π be a fragment circuit $\langle y_0, \ldots, y_k \rangle$ in G. In G^+, if $y_i \in X$, then y_{i-1} or y_{i+1} is in C_{sr}, by Lemma 10.1. By Lemma 10.3, every component of $G^+ - X$ except C_{sr} must intersect G_π. Lemma 10.5 implies that x_1x_2 is not an edge in G^+, where $X = \{x_1, x_2\}$.

DEFINITION 11.1. *Let $X = \{x_1, x_2\}$ be a 2-cut in G^+ such that $G^+ - X$ has exactly two components C_d and C_{nd}, where $X \cup V(C_d)$ induces a diamond $K_4 - x_1x_2$. If $C_d = C_{sr}$, then X is an sr-*diamond-cut*, whereas if $C_{nd} = C_{sr}$, then X is a* diamond-cut *(see Figure 6).*

We will prove that every 2-cut in G^+ is either a diamond-cut or an *sr*-diamond-cut, but not both. We begin by studying *sr*-diamond-cuts. The following lemma, which allows us to find a rooted $K_{2,2}$ given certain paths, is our main tool in the proof of

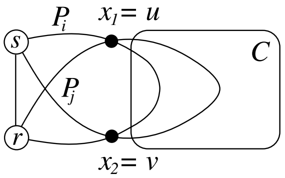


FIG. 7. *P_i and P_j are sr-paths in G^+.*

Theorem 11.3, which will allow us to characterize obstructions with *sr*-diamond-cuts. This lemma is essentially statement (2.1) of [16].

LEMMA 11.2. *Let G be a graph with distinguished vertices s, r, u, v. If G contains an s, u-path and an r, v-path that are disjoint, and G contains an s, v-path and an r, u-path that are disjoint, then G has as a rooted minor a $K_{2,2}$ in which the partite sets are $\{s, r\}$ and $\{u, v\}$.*

We use Lemma 11.2 to characterize *sr*-diamond-cuts in obstructions.

THEOREM 11.3. *Let G be an obstruction with fragment subgraph G_π. If X is a cut-set of size 2 in G^+, then X is an sr-diamond-cut if and only if $V(C_{sr}) \cap V(G_\pi) = \emptyset$.*

Proof. Let $X = \{x_1, x_2\}$ be the 2-cut in G^+. If X is an *sr*-diamond-cut, then $V(C_{sr}) = \{s, r\}$ by Definition 11.1. Since no fragment subgraph includes s or r, we obtain $V(C_{sr}) \cap V(G_\pi) = \emptyset$.

Conversely, suppose that $V(C_{sr}) \cap V(G_\pi) = \emptyset$. By Lemma 10.1, G_π cannot intersect X and must thus be entirely contained in some other component C of $G^+ - X$. By Lemma 10.3, C_{sr} and C are the only components of $G^+ - X$.

Fix a fragment structure $\langle P_1, \dots, P_k \rangle$ in G. Observe that every fragment path enters $V(C)$ from C_{sr} and returns to C_{sr}, so it must contain x_1 and x_2. If x_1 occurs before x_2 in every fragment path P_i, then contract the s, x_1-subpath of P_1 to s, contract the x_2, r-subpath of P_1 to r, and delete all remaining vertices and edges from C_{sr}. This is a proper *sr*-minor of G that contains the same fragment circuit, a contradiction. We argue similarly if x_2 precedes x_1 in every fragment path.

Now suppose that x_1 precedes x_2 in P_i and x_2 precedes x_1 in P_j. (See Figure 7.) Apply Lemma 11.2 to $G - V(C)$ with $u = x_1$, $v = x_2$, $P_{su} \cup P_{vr} = P_i - V(C)$, and $P_{sv} \cup P_{ur} = P_j - V(C)$. By Lemma 11.2, we can contract or delete all but four edges of $G - V(C)$ to obtain (as a subgraph) $K_{2,2}$ with roots $\{s, r, u, v\}$ arranged into partite sets $\{s, r\}$ and $\{u, v\}$. The new graph has a fragment structure for the same fragment circuit, obtained by replacing each fragment path P with the path $s, (P - V(C_{sr})), r$. By minor-minimality the rooted graph $K_{2,2}$ is a spanning subgraph of $G - V(C)$. Then $G - V(C)$ is $K_{2,2}$ by Lemma 10.6. Therefore $G^+ - V(C)$ induces a diamond $K_4 - x_1x_2$. □

Next we give a similar structural characterization of diamond-cuts. The following result of Duffin [8] on series-parallel graphs will be crucial as it plays a role in the proof of Theorem 11.5 similar to that of Lemma 11.2 in the proof of Theorem 11.3.

A graph H is a *series-parallel* graph if it can be obtained from any edge $e \in E(H)$ by a sequence of subdivisions and doublings of edges. Such operations cannot create a K_4-subdivision. In fact, a 2-connected graph is a series-parallel graph if and only if it contains no subdivision of K_4, as is implied by the following result of Duffin [8].

THEOREM 11.4. *If H is a 2-connected graph that contains no K_4-subdivision, and uv is an edge of H, then H can be obtained from uv by a sequence of subdivisions and doublings of edges.*

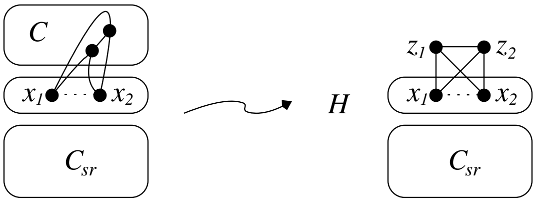


FIG. 8. *H, where C' contains a K_4-subdivision.*

Let H be a series-parallel graph that is built from the edge uv. As H is built we can label its vertices such that all u, v-paths are strictly increasing: start by labeling u with 0 and v with 1; when an edge is subdivided, give the new vertex a label that is the average of the labels of its 2 neighbors. We call this a *uv-numbering* of H.

THEOREM 11.5. *Let G be an obstruction with fragment subgraph G_π, If X is a cut-set of size 2 in G^+, then X is a diamond-cut if and only if $V(C_{sr}) \cap V(G_\pi) \neq \emptyset$.*

Proof. First assume that X is a diamond-cut. If $\pi = \langle y_0, y_1, y_2, \ldots, y_k\rangle$ is the fragment circuit, then since $|C_d| = 2$, but $|V(G_\pi)| \geq 3$ we see that $y_i \in X$ for some i if $V(C_{sr}) \cap V(G_\pi) = \emptyset$. Lemma 10.1 now yields $V(C_{sr}) \cap V(G_\pi) \neq \emptyset$.

For the converse, suppose that $V(C_{sr}) \cap V(G_\pi) \neq \emptyset$. Lemma 10.6 implies that $x_1x_2 \notin E(G)$. Let C be some component other than C_{sr} in $G^+ - X$, and let C' be the graph induced by $V(C) \cup X$ with the additional edge x_1x_2.

We observe that C' must be 2-connected, since any cut-vertex in C' would be a cut-vertex in G as well. We will first show that C' must contain a K_4-subdivision.

Suppose that C' contains no K_4-subdivision. Since $V(C_{sr}) \cap V(G_\pi) \neq \emptyset$, whenever the fragment circuit enters C (via X) it will eventually leave again (via X). We claim that it must enter and leave through different vertices of X. Consider an x_1x_2-numbering of C'. Any fragment path restricted to $V(C) \cup X$ is an x_1, x_2- or x_2, x_1-path, so within $V(C) \cup X$ its vertex labels are strictly increasing or strictly decreasing. Hence every fragment with vertices in $V(C) \cup X$ is strictly increasing or strictly decreasing. Therefore, if the fragment circuit enters C through x_1, then it must leave through x_2, and vice versa.

Let $H = G - V(C) + x_1x_2$. Since there is an x_1, x_2-path in $G[V(C) \cup X]$, H is an sr-minor of G. Let π' be the list of vertices that results from deleting the vertices of π that are in C. Since $x_1x_2 \in E(H)$ and π' has x_1 followed by x_2, π' is a closed walk in H. For each fragment path P_i that intersects $V(C)$, $P_i - V(C) + x_1x_2$ is an sr-path in H. If $y_{i+1}, \ldots, y_{j-1}$ is a maximal sublist of π in C, then $\{y_i, y_j\} = \{x_1, x_2\}$ and $y_{i-1}y_iy_j$ and $y_iy_jy_{j+1}$ are fragments in H with fragment paths $P_i - V(C) + x_1x_2$ and $P_j - V(C) + x_1x_2$, respectively. Thus π' is a fragment circuit in H, which contradicts the minor-minimality of the obstruction G.

From the contradiction we deduce that C' is 2-connected and contains a K_4-subdivision. Thus by Lemma 5.3 C' contains a rooted model of K_4 with roots $\{x_1, x_2\}$, and therefore C contains a rooted model of $K_4 - x_1x_2$. Contract the branch sets containing x_1 and x_2 to x_1 and x_2, respectively, and contract the other branch sets to (new vertices) z_1 and z_2. Let H' be the sr-graph that results from applying these contractions to $G^+[V(C_{sr}) \cup X \cup V(C)]$. (See Figure 8.) We wish to modify π to obtain a fragment circuit in H'. By minor-minimality, G^+ then equals H'. Since X is a diamond-cut in H', this suffices.

Removing vertices of $V(C_{sr}) \cup X$ from π breaks the closed walk into a set $\mathcal{S}$ of proper subwalks of π because $V(C_{sr}) \cap G_\pi \neq \emptyset$. Since π starts and ends in $V(C_{sr})$,

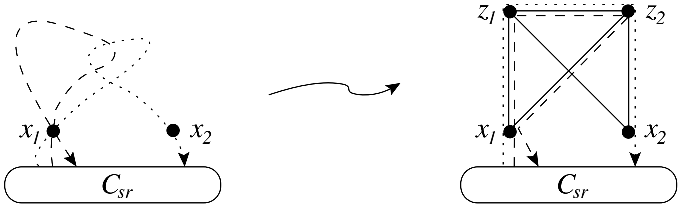


FIG. 9. *Replacements for walks that leave and reenter $X \cup V(C_{sr})$.*

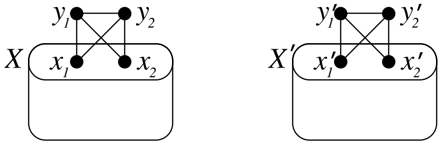


FIG. 10. *Two views of G^+, while showing that $\widehat{G}$ is well defined.*

each walk in $\mathcal{S}$ has the form $\langle y_{i+1}, \dots, y_{j-1}\rangle$ such that $y_i, y_j \in X$. Also, each walk in $\mathcal{S}$ is contained in a component of $G^+ - X$ other than C_{sr}. By Lemma 10.1, y_{i-1} and y_{i+1} are in C_{sr}. Let π' be the closed walk on H that we obtain by replacing each walk in $\mathcal{S}$ by $\langle z_1, z_2\rangle$. (See dotted or dashed lines in Figure 9.) Note that π' is a fragment circuit in H. □

12. Removing diamond-cuts. Theorems 11.3 and 11.5 show that every 2-cut in G^+ is either an *sr*-diamond-cut or a diamond-cut, but not both. We will now make use of this to form a new graph that does not have any 2-cuts.

DEFINITION 12.1. *Let G be an obstruction. For each cut-set $\{x_1, x_2\}$ of size* 2 *in G^+, replace $V(C_d)$ and its incident edges by the edge x_1x_2. Call x_1x_2 a* diamond edge. *If $\{x_1, x_2\}$ is an sr-diamond-cut, then also relabel $x_1 = s$ and $x_2 = r$. Let $\widehat{G}$ denote the resulting sr-graph. Let $\widehat{G}^-$ denote the graph $\widehat{G} - \{s, r\}$.*

Note that replacing a diamond with a diamond edge produces an *sr*-minor of G^+.

LEMMA 12.2. *$\widehat{G}$ is well defined (up to exchanging s and r if G^+ has an sr-diamond-cut).*

Proof. Let X be a 2-cut in G^+ with $X = \{x_1, x_2\}$. Let $X \cup Y$ be the vertex set of the resulting diamond, with $Y = \{y_1, y_2\}$. Similarly define $X' = \{x_1', x_2'\}$ and $Y' = \{y_1', y_2'\}$ for another 2-cut X'. (See Figure 10.) Assume that $X \neq X'$ or $Y \neq Y'$. Since $X = X'$ implies that $Y = Y'$, we may assume $X \neq X'$. It suffices to check that $(X \cup Y) \cap Y' = \emptyset$; then we can delete Y' and add $x_1'x_2'$ without affecting $G[X \cup Y]$. This allows us to replace $X' \cup Y'$ with $x_1'x_2'$ before we replace $X \cup Y$ with x_1x_2. Since X and X' are arbitrary 2-cuts, all the 2-cuts in G^+ can be dealt with in any order.

Suppose that $x_1 = y_1'$. Since $\{y_1, y_2\} \subseteq N(x_1) = N(y_1')$, we have $\{y_1, y_2\} \subseteq \{x_1', x_2', y_2'\}$. By symmetry, let $x_1' = y_1$. Now $N(x_1) \cap N(y_1) = N(y_1') \cap N(x_1')$ requires $y_2 = y_2'$. This yields $N(y_1) \cap N(y_2) = N(x_1') \cap N(y_2') = \{x_2'\}$, but $N(y_1) \cap N(y_2) = \{x_1, x_2\}$, which is a contradiction. Hence, $Y' \cap X = \emptyset$. Thus, if $(X \cup Y) \cap Y' \neq \emptyset$, then $Y \cap Y' \neq \emptyset$. Since $y_1'y_2'$ is an edge and X is a cut-set that doesn't intersect Y', we have $Y = Y'$. This requires $X = X'$, which we already excluded. Thus $(X \cup Y) \cap Y' = \emptyset$, and $\widehat{G}$ is well defined. □

The following example illustrates Definition 12.1.

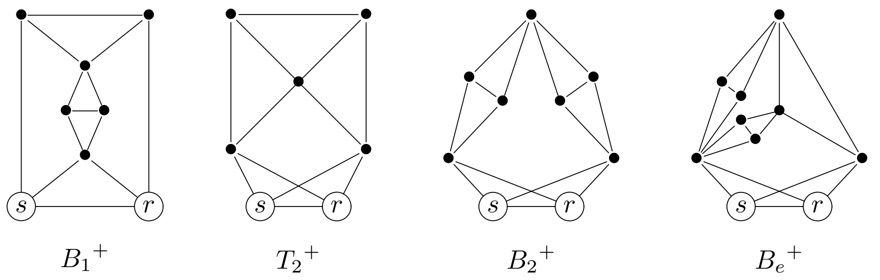


FIG. 11. *Examples of G^+ with* 2*-cuts.*

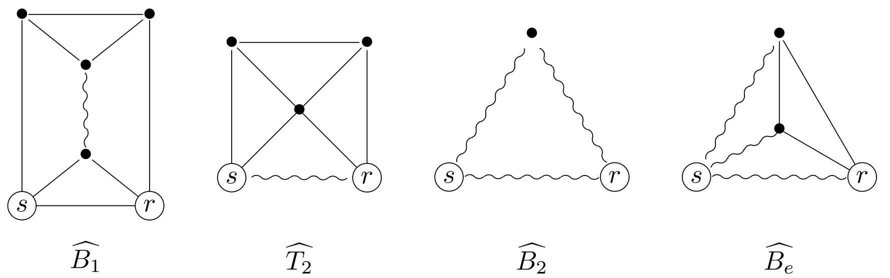


FIG. 12. *Examples of $\widehat{G}$ with diamond edges.*

Example 12.3. Consider all the graphs G from $\mathcal{F}_0'$ so that G^+ has a 2-cut. Figure 11 shows G^+ for these graphs, and Figure 12 shows the corresponding $\widehat{G}$ with the diamond edges drawn as wavy lines. Note that for the other *sr*-graphs $G \in \mathcal{F}_0'$, G^+ does not have a 2-cut, so that $\widehat{G} = G^+$ for these.

B_1^+ has one diamond-cut, and $\widehat{B_1}$ is a triangular prism with s and r in one triangle. The diamond edge connects the third vertex on the triangle with the other triangle.

T_2^+ has an *sr*-diamond-cut, and $\widehat{T_2}$ is an *sr*-rooted wheel on five vertices, obtained by joining one vertex to every vertex on a 4-cycle in which s and r are consecutive vertices. Here sr is the lone diamond edge.

$\widehat{B_2}$ is the *sr*-rooted triangle in which every edge is a diamond edge, and $\widehat{B_e}$ is the *sr*-rooted complete graph on four vertices in which the diamond edges are the edges incident with s (or r, depending on how we contract the *sr*-diamond-cut).

The next remark collects some observations on the structure of $\widehat{G}$.

Remark 12.4. It is easy to see that $\widehat{G}$ is an *sr*-minor of G^+. Every vertex of $\widehat{G}$ is also a vertex of G^+, except that it may have a new name s or r when sr is a diamond edge.

Suppose that uv is a diamond edge in a graph H. To reverse the operation of creating uv from a diamond, identify $\{u, v\}$ with two vertices of a copy of K_4 then deleting the edge uv; let H' be the resulting graph. If X is a cut-set of H and $C_1, \ldots, C_k$ are the components of $H - X$, then $\{u, v\}$ intersects at most one component of $H - X$, so X is a cut-set of H' and $C_1, \ldots, C_k$ are contained in k distinct components of $H' - X$. Since G^+ can be obtained from $\widehat{G}$ by a series of such

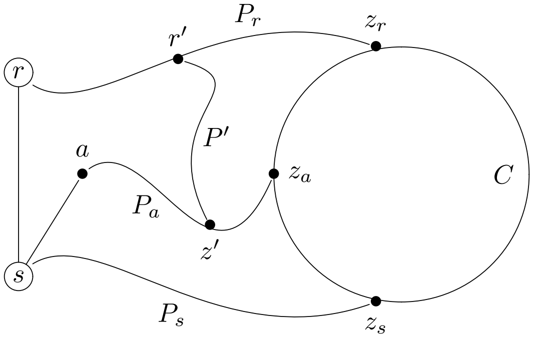


FIG. 13. *A model of $\widehat{T_2}$ in $\widehat{G}$.*

operations, if X is a cut-set in $\widehat{G}$ and $C_1, \dots, C_k$ are the components of $\widehat{G} - X$, then X is a cut-set in G^+ and $C_1, \dots, C_k$ are contained in distinct components of $G^+ - X$.

Since every cut-set in $\widehat{G}$ is also a cut-set in G^+ and the 2-cuts in G^+ are not 2-cuts in $\widehat{G}$, it follows that $\widehat{G}$ has no cut-sets of size at most 2. Thus $\widehat{G}$ is 3-connected, unless it is a triangle. This implies that the graph $\widehat{G}^-$ obtained from $\widehat{G}$ by removing s and r is connected.

If G^+ has no *sr*-diamond-cut, then G may be similarly obtained from $\widehat{G} - sr$ by series of these reverse operations; hence every cut-set of $\widehat{G} - sr$ is also a cut-set of G that breaks $G - X$ up into components as in $\widehat{G} - sr$. If G^+ has no *sr*-diamond-cut and X is a 2-cut in G, then $C_s \neq C_r$ in $G - X$ if and only if sr is a cut-edge in $\widehat{G} - X$, because each statement is true if and only if sr is a cut-edge in $G^+ - X$.

We will now reduce to the case when G^+ has no *sr*-diamond-cut.

THEOREM 12.5. *Let G be an obstruction. If G^+ has an sr-diamond-cut, then $G \in \{T_2, B_2, B_e\}$.*

Proof. Let $X = \{x_1, x_2\}$ be the *sr*-diamond-cut. Since X is also a cut-set in G (and in $G - X$ we have $C_s = \{s\}$ and $C_r = \{r\}$) it follows from Lemma 10.2 that G_π is entirely contained in C_{nd}.

Case 1. $\widehat{G}^-$ contains a cycle C. We show that $G = T_2$.

Since sr is a diamond edge, by Remark 12.4 it suffices to show that $\widehat{T_2}$ (see Figures 12 and 13) is an *sr*-minor of $\widehat{G}$. This implies that T_2 is an *sr*-minor of G, and thus by minor-minimality $G = T_2$.

Let $a \in V(\widehat{G}^-)$ be a neighbor of either s or r in $\widehat{G}$; by symmetry we may assume that $a \in N_{\widehat{G}}(s)$. Since $\widehat{G}^-$ has more than one vertex, $\widehat{G}$ is not a triangle, so $\widehat{G}$ is 3-connected. Hence there are pairwise disjoint paths P_a, P_s, P_r in $\widehat{G}$ from a, s, r to some vertices $z_a, z_s, z_r \in V(C)$, respectively. By choosing paths of minimal length, each path intersects $V(C)$ only at its endpoint. (The path P_a could be trivial.)

Since $\widehat{G}$ has no 2-cut, there is a shortest path P' from $V(P_r) - z_r$ to $V(C) \cup V(P_a) \cup V(P_s) - \{z_r, s\}$ in $\widehat{G} - \{z_r, s\}$. Let r' be the endpoint of P' in $V(P_r) - z_r$, and let z' be the other endpoint of P'. We perform the following contractions to obtain the desired $\widehat{T_2}$. Contract the r, r'-subpath of P_r into r. Contract z' into either z_a or z_s, depending on whether z' is in $V(P_a)$, $V(P_s) - s$, or $C - z_r$. Thus we obtain a subdivision of $\widehat{T_2}$ with center z' and 4-cycle s, r, z_r, z_s (if $z' = z_a$) or s, r, z_r, z_a (if $z' = z_s$).

Case 2. $\widehat{G}^-$ is acyclic and hence is a tree. We show that G is either B_2 or B_e.

Since $\widehat{G}$ has no cut-sets of size at most 2, every leaf of $\widehat{G}^-$ is adjacent to both s

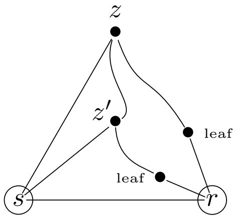


FIG. 14. *The case that* $sz, sz' \in E(\widehat{G})$.

and r. We first show that $\widehat{G}$ has at least two diamond edges other than sr. Observe that in $G-X$ we have $C_s = \{s\}$ and $C_r = \{r\}$, so by Lemma 10.1 the fragment circuit π cannot intersect X and is contained in $G^- - X$. Since the minimum degree of G_π is at least 2, $G^- - X$ is not a tree, so $\widehat{G}$ contains at least one diamond edge other than sr. Moreover, by Theorem 11.5, G_π is not entirely contained in the diamond of this cut. Since G_π has minimum degree 2 and $\widehat{G}^-$ is a tree, there must be at least two diamond edges other than sr.

If two such diamond edges lie on a path from s to r in $\widehat{G}$, then we obtain B_2 as an sr-minor of G (see Figure 12). Then minor-minimality implies $G = B_2$. If e and e' are diamond edges in $\widehat{G}$ not incident to r and s, respectively, then $\widehat{G}^-$ contains a unique path between e and e', and this path extends via two leaves in $\widehat{G}^-$ to an sr-path in $\widehat{G}$ that contains e and e'. Therefore we may assume that all diamond edges are incident to s or all diamond edges are incident to r. By symmetry, we will assume that all diamond edges are incident to s. If two such diamond edges are sz and sz', where z and z' are vertices of the tree $\widehat{G}^-$, then there is a path from z to z' in $\widehat{G}^-$ that extends to leaves in each direction. These leaves are adjacent to r, since every leaf is adjacent to both s and r. (See Figure 14.) Contract the paths from the two leaves to z and z'. This yields a K_4-subdivision in $\widehat{G}$, with branch vertices s, r, z, z' such that the edges incident to s are all diamond edges. Thus G has B_e as an sr-minor as described in Example 12.3. By minor-minimality, we conclude that $G = B_e$. □

Thus from now on we can assume that if G is an obstruction, then every 2-cut in G^+ is a diamond-cut. In Figure 12 we can see that B_1 is the only sr-graph in $\mathcal{F}_0'$ such that B_1^+ has a diamond-cut but no sr-diamond-cut.

13. Tools for 3-cuts in $\widehat{G}$.

From Remark 12.4, we may assume that $\widehat{G}$ is 3-connected or a triangle. In this section, we first study those 3-cuts in $\widehat{G}$ that have the form $\{s, r, x\}$. A cut-set in $\widehat{G}$ is also a cut-set in G^+, so if $\{s, r, x\}$ is a cut-set in $\widehat{G}$, then x is a cut-vertex in G^-. The next lemma studies cut-vertices in G^-.

LEMMA 13.1. *Let G be an obstruction.*

(i) *If x is a cut-vertex of G^-, then for each component C of $G - \{s, r, x\}$, there are s, x- and r, x-paths with more than two vertices that have all their internal vertices in C.*

(ii) *If x is a cut-vertex of G^-, then $sx \notin E(G)$ and $rx \notin E(G)$.*

(iii) *G^- has no cut-edge.*

Proof. (i) Let C be a component of $G - \{s, r, x\}$. By Lemma 10.4, none of the 2-subsets of $\{s, r, x\}$ is a cut-set of G, so each member of $\{s, r, x\}$ has a neighbor in C. Since C is connected, we get the desired paths.

For the proof of (ii) and (iii), let $\langle y_0, \dots, y_k\rangle$ be a fragment circuit with fragment structure $\langle P_1, \dots, P_k\rangle$ in G.

(ii) Suppose that P_i is a fragment path that contains sx. Since sx must be the first edge of P_i, it follows that $P_i - \{s, r, x\}$ is contained in a single component of $G - \{s, r, x\}$ and sx is not in $y_{i-1}y_iy_{i+1}$ (in either direction). By (i) every other component of $G - \{s, r, x\}$ contains an s, x-path. Modify P_i by replacing sx with one such s, x-path; this is an sr-path in $G - sx$ that contains $y_{i-1}y_iy_{i+1}$. Since we can do this for every fragment path, $sx \notin E(G)$ by minor-minimality. Similarly, $rx \notin E(G)$.

(iii) Suppose that e is a cut-edge of G^- with endpoints $\{u, v\}$. Contract e in G. We modify the fragment circuit and fragment structure in G to obtain a fragment circuit and structure in $G \cdot e$.

First we convert sr-paths in G to sr-paths in $G \cdot e$. Consider an sr-path P. The edges of P continue to form an sr-path in $G \cdot e$ unless P contains the edge e, in which case $P \cdot e$ is an sr-path in $G \cdot e$.

For all $1 \le i \le k$ such that P_i contains e and $e \notin \{y_{i-1}y_i, y_iy_{i+1}\}$, replace P_i in the fragment structure by $P_i \cdot e$.

If $e = y_iy_{i+1}$ for some $1 \le i \le k$, replace y_i, y_{i+1} in the fragment circuit by the new vertex e, and replace P_i, P_{i+1} in the fragment structure by $P_i' \cdot e$, where P_i' is the sr-path in G formed by concatenating the s, y_i-portion of P_i and the y_{i+1}, r-portion of P_{i+1}. This P_i' will be a fragment path in $G \cdot e$, and the subpath y_{i-1}, e, y_{i+2} will be its fragment. Since P_i contains y_{i-1}, y_i, y_{i+1} and P_{i+1} contains y_i, y_{i+1}, y_{i+2}, we have $e \notin \{y_{i-1}y_i, y_{i+1}y_{i+2}\}$. Thus we can replace y_i, y_{i+1} and P_i, P_{i+1} independently for all $1 \le i \le k$ such that $e = y_iy_{i+1}$.

We have produced a fragment circuit with a fragment structure in $G \cdot e$, contradicting the minor-minimality of the obstruction G. □

Using Lemma 13.1, we now show that if $\widehat{G}$ has a 3-cut of the form $\{s, r, x\}$, then it is one of the graphs from $\mathcal{F}_0'$. Thus in our search for obstructions that are not in $\mathcal{F}_0'$, we will be able to assume that G^- is 2-connected.

THEOREM 13.2. *If G is an obstruction and G^- has a cut-vertex, then $G \in \{B_0, B_1, B_2\}$.*

Proof. Let x be a cut-vertex of G^-, and let $X = \{s, r, x\}$. If G^+ has an sr-diamond-cut, then by Theorem 12.5 it follows that $G = B_2$, since $T_2 - \{s, r\}$ and $B_e - \{s, r\}$ are 2-connected. Thus we may assume that every 2-cut in G^+ is a diamond-cut.

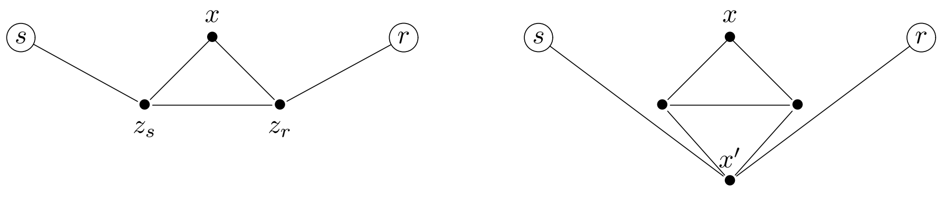
Let B be a block of G^- that contains x, let C be the component of $G - X$ that contains $B - x$, and let L be $G[V(C) \cup X]$. By Lemma 13.1, B is not an edge, so B is 2-connected and B has at least 3 vertices.

Case 1. There are two disjoint paths in $L - x$ from $B - x$ to $\{s, r\}$.

Consider minimal such paths and let z_s, z_r be their endpoints in $B - x$. Since B is 2-connected, it contains internally disjoint paths P_s, P_r that connect z_s and z_r to x, and these paths are connected by some path in $B - x$. Contracting $P_s - x$ into z_s and $P_r - x$ into z_r, it can be seen that L has the rooted minor with roots $\{s, r, x\}$ in Figure 15, Case 1.

Case 2. There are no two disjoint paths in $L - x$ from $B - x$ to $\{s, r\}$.

By Lemma 13.1(i), s and r each have a neighbor in $L - X$. Hence the absence of disjoint paths from $B - x$ to $\{s, r\}$ in $L - x$ requires a cut-vertex x' of $L - x$ that separates $\{s, r\}$ from $B - x$. Since $L - X$ is connected, x' is not s or r. By Lemma 13.1(iii), $B - \{x, x'\}$ is nonempty, so $\{x, x'\}$ is a cut-set in G that separates $B - \{x, x'\}$ from $\{s, r\}$. Therefore $\{x, x'\}$ is a diamond-cut. Again L contains paths



Case 1 Case 2

FIG. 15.

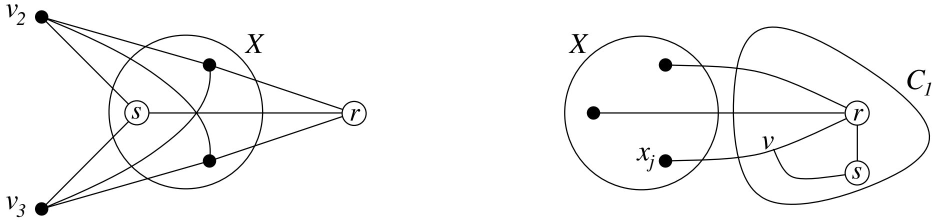


FIG. 16. *If $s \in X$, and if $s \notin X$.*

from B to s and to r. By Lemma 13.1(ii), these paths must contain x'. Therefore L has the rooted minor with roots $\{s, r, x\}$ in Figure 15, Case 2.

Now consider two blocks of G^- that each contain x, and also consider the two components of $G - X$ that contain these blocks. If for both components we have Case 1, then G has the sr-minor B_0. If we have two Case 2s, then G has the sr-minor B_2. If we have one Case 1 and one Case 2, then G has the sr-minor B_1. Finally, by minor-minimality of the obstruction, G is isomorphic to one of these sr-minors. □

If $V(\widehat{G}) = \{s, r, x\}$ and sr is not a diamond edge, then $\widehat{G} - sr$ is a path with three vertices and x is a cut-vertex. By Remark 12.4, x is a cut-vertex in G, contradicting the fact that obstructions are 2-connected. Therefore we may assume that $\widehat{G}$ is not a triangle, and hence $\widehat{G}$ is 3-connected.

Theorem 13.2 takes care of any 3-cuts in $\widehat{G}$ of the form $\{s, r, x\}$. The next theorem shows that any other 3-cut in $\widehat{G}$ cannot break the graph up into too many components.

THEOREM 13.3. *If G is an obstruction and $\widehat{G}$ has a cut-set X of size* 3 *with $\{s, r\} \not\subseteq X$ such that $\widehat{G} - X$ has at least* 3 *components, then $G = F_0$.*

Proof. Let X be a 3-cut in $\widehat{G}$ with $\{s, r\} \not\subseteq X$. Without loss of generality, we may assume that $r \notin X$.

Let C_1, C_2, and C_3 be three components of $\widehat{G} - X$, with $r \in V(C_1)$. Choose $v_2 \in V(C_2)$ and $v_3 \in V(C_3)$. Since $\widehat{G}$ is 3-connected, for every vertex $v \in V(C_i)$ we can find a set $\mathcal{P}(v)$ of three internally disjoint paths that start at v and are entirely contained in C_i except for their endpoints, which are the three vertices of X. If $s \in X$, then $\mathcal{P}(r), \mathcal{P}(v_2), \mathcal{P}(v_3)$ yield the desired F_0^+-minor in $\widehat{G}$, so $G = F_0$. (See Figure 16.) If on the other hand $s \notin X$, then $s \in V(C_1)$ since s and r are adjacent in $\widehat{G}$. Consider a path P_s from s to X in $\widehat{G} - r$; it first meets a path $P_r \in \mathcal{P}(r)$ at a vertex v. Contracting the s, v-portion of P_s and the v, X-portion of P_r identifies s with a vertex of X, and then we can argue as in the previous case. □

We will now specify the properties that any remaining obstructions must possess.

Definition 13.4. *If G is an obstruction such that G^+ has no sr-diamond-cut, G^- (and thus $\widehat{G}^-$) is 2-connected, $\widehat{G}$ is 3-connected, $\widehat{G}$ has tree-width at most 3, and there is no cut-set of size 3 that breaks $\widehat{G}$ into more than two components, then we say that G is* relevant.

We have shown so far that if an obstruction is not relevant, then it must be T_2, B_2, B_e (Theorem 12.5) or B_0, B_1, B_2 (Theorem 13.2 and Remark 7.6) or T_0 (Theorem 9.7 and $\widehat{G}$ is a minor of G^+) or F_0 (Theorem 13.3). Therefore, to finish the proof of Theorem 8.2, it suffices to show that a relevant obstruction must also be in $\mathcal{F}_0'$. To hunt down the remaining minors T_s, T_r and F_1, we now need to study tree-decompositions of relevant obstructions.

14. Tree-decompositions. In this section, we give some of the standard terminology and basic results on tree-decompositions that we will use in the rest of the paper. For a more thorough introduction to tree-decompositions, see Diestel [7] and the survey of Bodlaender [5] .

Definition 14.1. *Let G be a graph, let T be a tree, and let $\mathcal{V} = \{V_t : t \in V(T)\}$ be a family of subsets of the vertices of G indexed by the vertices of T. The pair $(T, \mathcal{V})$ is a* tree-decomposition *of G if it satisfies the following conditions:*

1. *Every vertex of G is in some V_t.*
2. *Every edge in G belongs to $G[V_t]$ for some $t \in T$.*
3. *For every $u \in V(G)$, the set $T(u)$ induces a subtree of T, where $T(u) = \{t \in V(T) : u \in V_t\}$.*

The width *of a tree-decomposition is the maximum of $|V_t| - 1$ over $t \in V(T)$, and the* tree-width *of G is the minimum width among all tree-decompositions of G. We refer to the sets V_t in a tree-decomposition as* node-sets.

Every nontrivial tree has tree-width 1. In general, the smaller the size of the node-sets, the more closely G resembles a tree. The tree-width is thus a measure of how tree-like the graph is. An important feature of a tree-decomposition is that it transfers the separation properties of its tree to the graph decomposed. We summarize some of these basic properties.

Remark 14.2. Let $(T, \mathcal{V})$ be a tree-decomposition of a graph G.

1. Property 14.1.3 is equivalent to the following statement: if t_2 is on the path from t_1 to t_3 in T, then $V_{t_1} \cap V_{t_3} \subseteq V_{t_2}$.
2. If t_1 and t_2 are adjacent vertices in T, then $G - (V_{t_1} \cap V_{t_2})$ is the disjoint union of two subgraphs; one subgraph is induced by $\bigcup_{t \in V(T_1)} V_t - V_{t_2}$ and the other is induced by $\bigcup_{t \in V(T_2)} V_t - V_{t_1}$, where T_1 and T_2 are the components of $T - t_1t_2$ that contain t_1 and t_2, respectively.
3. If t is a vertex of T, and $T_1, \ldots, T_k$ are the components of $T - t$, then $G - V_t$ is the disjoint union of the k subgraphs $G[U_i]$, where $U_i = \bigcup_{t' \in V(T_i)} V_{t'} - V_t$.

The observations above give us information about the graph when we know that there is a path in the tree. The next lemma tells us about the structure of the tree, given a path in the graph. We first define a special type of path that we are interested in.

Definition 14.3. *Let G be a graph, and let S be a set of vertices in G. If P is a path in G with at least three vertices whose endpoints are both in S but none of whose internal vertices are in S, then we say that P is* S-external.

If the graph has an S-external path where S corresponds to a node-set in a tree-decomposition, then we can prove the following about the structure of the host tree.

Lemma 14.4. *Let $(T, \mathcal{V})$ be a tree-decomposition of a graph G. If $a, b \in V_t$ for some $t \in V(T)$ and P_{ab} is a V_t-external a, b-path, then there is a neighbor t' of t*

in T such that $V_{t'}$ contains a and b and for every internal vertex u of P_{ab}, $T(u)$ is contained in the component of $T - t$ containing t'.

Proof. Let P'_{ab} be the path $P_{ab} - \{a, b\}$, and let T_i and U_i be as in Remark 14.2.3. Since $G - V_t$ is the disjoint union of the $G[U_i]$'s, P'_{ab} is contained in $G[U_i]$ for some i. Hence for every vertex $u \in V(P'_{ab})$, the subtree $T(u)$ is contained in T_i. If ax is the first edge of P_{ab}, then $\{a, x\}$ is contained in some node-set $V_{t_x} \in \mathcal{V}$. Since $x \in V(P'_{ab})$, we have $t_x \in V(T_i)$. Let t' be the neighbor of t in T_i; t' is on the t, t_x-path in T, so by Remark 14.1.1, $a \in V_{t'}$. Similarly, $b \in V_{t'}$. □

For our investigations we require a particular structure of our tree-decomposition that is described below.

DEFINITION 14.5. *A tree-decomposition $(T, \mathcal{V})$ of a graph G is called* k-alternating *if it satisfies the following properties:*

1. *The node-sets are distinct sets of sizes k and $k + 1$ only.*
2. *Every edge of T joins node-sets of unequal sizes.*
3. *If $t_1t_2 \in E(T)$ and $|V_{t_1}| = k$, then $V_{t_1} \subset V_{t_2}$.*
4. *the node-set corresponding to a leaf of T has size $k + 1$.*

THEOREM 14.6. *If G has tree-width k, then G has a k-alternating tree-decomposition.*

Proof. Start from an arbitrary tree-decomposition $(T, \mathcal{V})$ of width k. First obtain a tree-decomposition in which all node-sets are of size $k + 1$ and the intersection of two node-sets whose vertices are adjacent in T has size k, by repeating the following operations until no such operation is possible (see [5, 6]):

- If $tt' \in E(T)$ and $V_t \subseteq V_{t'}$, then contract the edge tt' to t', deleting V_t from $\mathcal{V}$.
- If $tt' \in E(T)$, $|V_t| < k + 1$, and $V_{t'} \not\subseteq V_t$, then add a vertex $v \in V_{t'} - V_t$ to V_t.
- If $tt' \in E(T)$, $|V_t| = |V_{t'}| = k + 1$, and $|V_t \cap V_{t'}| < k$, then choose vertices $v \in V_t - V_{t'}$ and $v' \in V_{t'} - V_t$, and subdivide the edge tt' with a vertex t'' such that $V_{t''} = V_t - v + v'$.

Once we have found this tree-decomposition of G, we subdivide every edge $tt' \in E(T)$ with a vertex t'' such that $V_{t''} = V_t \cap V_{t'}$. This tree-decomposition will satisfy properties 2 through 4. To show that the first property holds, we may have to modify the decomposition so that all node-sets are distinct. By Remark 14.2.1, two identical node-sets $V_t = V_{t'}$ must have size k.

Consider the tree obtained by deleting the first edge on the t, t'-path, adding the edge tt' and then contracting it, thus identifying V_t and $V_{t'}$. This yields a tree-decomposition of G with a smaller tree, still satisfying properties 2 through 4. Repeating this operation until all node-sets of $\mathcal{V}$ are distinct produces a tree-decomposition $(T, \mathcal{V})$ that satisfies properties 1–4. □

We conclude this section with some simple observations about k-alternating tree-decompositions.

Remark 14.7. Let $(T, \mathcal{V})$ be a k-alternating tree-decomposition of G.

1. Remark 14.2.3 implies that if G is a connected graph and V_t is a node-set that corresponds to a nonleaf vertex t in T, then V_t is a cut-set and removing V_t breaks G into vertex-disjoint subgraphs that correspond to the components of $T - t$. (These subgraphs are not necessarily connected; each may contain more than one component of the remaining graph.)
2. No node-set of size $k + 1$ can be a minimal cut-set in G, because it contains a size k node-set that corresponds to a neighboring vertex in the tree.

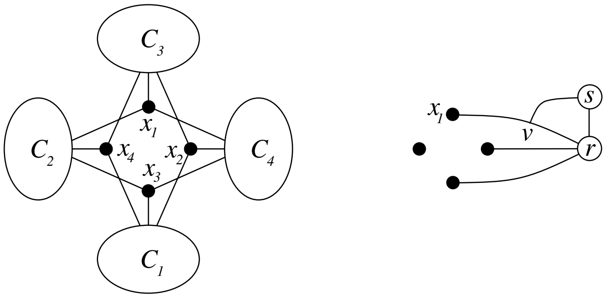


FIG. 17. *Proof of Lemma* 15.2 *if* $s \in X$*, if* $s \notin X$.

3. If V_t is a node-set of size $k+1$ and $V_{t'}$ is a node-set of size k such that $V_{t'} \subset V_t$, then $tt' \in E(T)$: since $V_{t'} \subset V_t$, it follows from Remark 14.2.1 that $V_{t'}$ is contained in the node-set corresponding to each vertex on the path from t to t'. If $tt' \notin E(T)$, then there is a vertex $t'' \in T$ on the path from t to t' whose corresponding node-set has size k, but this implies that $V_{t''} = V_{t'}$, a contradiction.

15. 3-alternating tree-decompositions of $\widehat{G}$.

We concluded section 13 by identifying properties of the obstructions we have not yet characterized. If G is a relevant obstruction, then Definition 13.4 tells us that $\widehat{G}$ has tree-width exactly 3. Thus, we may apply Theorem 14.6 to $\widehat{G}$ and obtain a 3-alternating tree-decomposition of $\widehat{G}$. In this section, we use the fact that G is a relevant obstruction to establish some further properties of such a tree-decomposition of $\widehat{G}$.

LEMMA 15.1. *Let* $(T, \mathcal{V})$ *be a* 3*-alternating tree-decomposition of* $\widehat{G}$*, where* G *is a relevant obstruction. If* $|V_t| = 3$*, then* t *has degree* 2 *in* T.

Proof. By Definition 14.5.4, the degree of t is at least two. By Remark 14.7.1, removing V_t breaks $\widehat{G}$ into at least as many components as are in $T - t$, i.e., at least $d_T(t)$. However, by the definition of a relevant obstruction, $\widehat{G} - V_t$ has at most two components. Hence the degree of t is exactly 2. □

It also holds that the maximum degree in the host tree T for a 3-alternating tree-decomposition of $\widehat{G}$ is at most 3. This follows from the next lemma and the fact that, by Remark 14.7.1, every 3-set of $\mathcal{V}$ is a cut-set of $\widehat{G}$.

LEMMA 15.2. *Let* G *be an obstruction. If* $X \subset V(\widehat{G})$ *is a* 4*-set, then one of its subsets of size* 3 *is not a cut-set.*

Proof. Let $X = \{x_1, x_2, x_3, x_4\}$, and suppose that every set $X_i = X - x_i$ is a cut-set in $\widehat{G}$. If $\{s, r\} \subset X$, then Theorem 13.2 implies that $G \in \{B_0, B_1, B_2\}$. For each element of this set, the claim is easily checked. Hence we may assume that $r \notin X$ (the case $s \notin X$ is similar).

Let C_i be a component of $\widehat{G} - X_i$ that does not contain x_i but, if possible, contains r. Each C_i is a component of $\widehat{G} - X$. Since $\widehat{G}$ is 3-connected, each vertex in X_i has a neighbor in C_i. Thus $C_1, \dots, C_4$ are distinct and pairwise disjoint. (See Figure 17.) Contracting each C_i to a vertex y_i yields Q_3 in which one partite set is X and where x_j and y_j are antipodal vertices. Thus if $s \in X$ and r is in some C_i, then G^+ has Q_3 as an *sr*-minor with s and r adjacent, which contradicts Corollary 9.3. If, however, $s \in X$ and r is in some other component C of $\widehat{G} - X$, then contracting C to a new vertex labeled r we see that r is adjacent to every vertex in X and replacing y_1 by r we get another contradiction to Corollary 9.3.

Now suppose that $s \notin X$. Thus s and r are in the same component C of $\widehat{G} - X$. By 3-connectedness there must be three internally disjoint r, X-paths, say, P_1, P_2, P_3 ending in x_1, x_2, x_3, respectively. By 3-connectedness there must also be an s, X-path P that avoids r and intersects one of the P_i (if only in x_i). Let v be the first vertex on P that is on any P_i, say P_1. Contract the s, v-subpath of P and the v, x_1-subpath of P_1 (this identifies s with x_1); contract the rest of C to r to obtain a new vertex that plays the role of y_4 in the Q_3-minor. This produces the previous case as a rooted minor, so again G^+ has an sr-minor that contradicts Corollary 9.3. □

To conclude the section, we locate the node-set(s) containing s and r.

Lemma 15.3. *Let $(T, \mathcal{V})$ be a 3-alternating tree-decomposition of $\widehat{G}$, where G is a relevant obstruction. There is exactly one node $t \in V(T)$ whose node-set V_t contains both s and r, and t is a leaf of T.*

Proof. Since sr is an edge of $\widehat{G}$, by Definition 14.1 there is at least one vertex of T whose node-set contains both s and r. Furthermore, such node-sets cannot be of size 3, since these node-sets are cuts (by Remark 14.7.1) and that would imply that G^- has a cut-vertex, contradicting G being a relevant obstruction.

Now consider the subtree induced by all the vertices of T whose node-sets contain both s and r. Since all these node-sets must be of size 4, it follows from Definition 14.5.3 that the subtree has order one. Hence there is only one node-set in $\mathcal{V}$ that contains both s and r.

Let t be the vertex of T whose corresponding node-set V_t is $\{s, r, u, v\}$. Since s, r are together in only one node-set and the node-sets corresponding to the neighbors of t in T must be distinct 3-element subsets of V_t by Definition 14.5, t has at most two neighbors in T. Suppose that t has neighbors t_r and t_s in T. We may write $V_{t_s} = \{s, u, v\}$ and $V_{t_r} = \{r, u, v\}$.

If there is a V_t-external sr-path in $\widehat{G}$, then we get a contradiction to Lemma 14.4. Since G is a relevant obstruction, there is no sr-diamond-cut. Hence every V_t-external sr-path in G^+ corresponds to a V_t-external sr-path in $\widehat{G}$. We conclude that there is no V_t-external sr-path in G^+. Hence there is no sr-path in $G - \{u, v\}$, implying that $C_s \neq C_r$ in $G - \{u, v\}$.

By Definition 14.5.4, neither t_s nor t_r is a leaf of T. Let t'_s and t'_r be neighbors of t_s and t_r other than t, respectively, and let x and y be vertices in $V_{t'_s} - V_{t_s}$ and $V_{t'_r} - V_{t_r}$, respectively. By 3-connectedness $\widehat{G}$ has an x, y-path P avoiding $\{u, v\}$. By Remark 14.7.1 V_{t_s} is a cut-set that separates x from y, so P contains s. Also V_{t_r} separates x and s from y, so P contains r. This contradicts Lemma 10.5. Thus, t must have degree less than two in T. Since $\widehat{G}^-$ is 2-connected, $\widehat{G}$ has at least five vertices, so that T cannot be K_1. Hence t is a leaf of T. □

Hence, in addition to the properties of a 3-alternating tree-decomposition specified in Definition 14.5 and Remark 14.7, a 3-alternating tree-decomposition of $\widehat{G}$ when G is a relevant obstruction has the following properties:

1. every vertex of the tree whose node-set has size 3 is of degree two,
2. every vertex of the tree whose node-set has size 4 is of degree at most three, and
3. s and r are in exactly one node-set, which corresponds to a leaf in the tree T.

Remark 15.4. We reformulate Lemma 14.4 for the context in which we will use it, as follows.

Let $(T, \mathcal{V})$ be a 3-alternating tree-decomposition of $\widehat{G}$, where G is a relevant obstruction. Let t be a vertex of T with $|V_t| = 4$, and suppose that P is a V_t-external

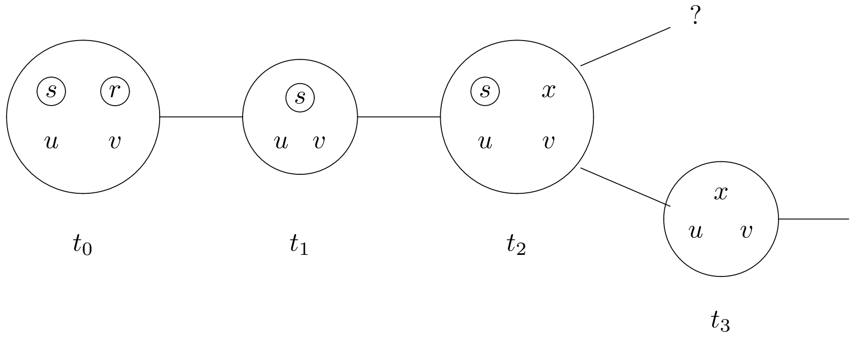


FIG. 18. *Tree decomposition of $\widehat{G}$.*

v_1, v_2-path in $\widehat{G}$. Then t has a neighbor t' such that $\{v_1, v_2\} \subset V_{t'} \subset V_t$, and for every internal vertex u of P, $T(u)$ is contained in the component of $T - t$ that contains t'.

16. Exploring a 3-alternating tree-decomposition of $\widehat{G}$.

We now explore the 3-alternating tree-decomposition of $\widehat{G}$ starting from the leaf whose node-set contains s and r. For the rest of this section, we fix $(T, \mathcal{V})$ to be a 3-alternating tree-decomposition of $\widehat{G}$ (where G is a relevant obstruction) with a leaf t_0 such that $V_{t_0} = \{s, r, u, v\}$. Let t_1 be the neighbor of t_0 in T. A node-set corresponding to t_1 can only be $\{s, u, v\}$ or $\{r, u, v\}$.

So far, the assumptions we have made in our proofs have not distinguished s from r. In this section, we break from this pattern by assuming that t_0 has a neighbor t_1 with corresponding node-set $\{s, u, v\}$. We will prove that in this case G must be F_1 or T_s. Switching s and r in these graphs leaves F_1 unchanged but changes T_s into T_r. Therefore, if we assume that t_0 has a neighbor with node-set $\{r, u, v\}$ instead, then switching s and r in the proofs that follow will imply that G must be F_1 or T_r. This completes the proof that that every graph $G \in \mathcal{F}_0$ has some graph from $\mathcal{F}_0'$ as an sr-minor.

Since V_{t_0} is the only node-set containing r, we have $N_{\widehat{G}}(r) \subseteq \{s, u, v\}$. Since $\widehat{G}$ is 3-connected, equality holds. Hence $\{u, v\}$ is a 2-cut in G, so Lemma 10.6 implies that $uv \notin E(G)$. Furthermore, $\{u, v\}$ is not a diamond-cut in G^+, since otherwise by applying Lemma 10.3 the diamond component would have to intersect G_π, and applying Lemma 10.2 would contradict Theorem 11.5. Thus uv is not a diamond-edge, so $uv \notin E(\widehat{G})$.

Since $V_{t_1} = \{s, u, v\}$, it follows that t_1 has a neighbor t_2 with $V_{t_2} = \{s, u, v, x\}$ for some vertex x. By Lemma 15.1, t_1 has no other neighbors. The next lemma establishes that t_2 has a neighbor t_3 with $V_{t_3} = \{u, v, x\}$.

LEMMA 16.1. *In the tree-decomposition of $\widehat{G}$, there is a neighbor of t_2 whose corresponding node-set is $\{u, v, x\}$.*

Proof. Since $\widehat{G}^-$ is 2-connected, $\{s, r, x\}$ is not a cut-set of $\widehat{G}$. Let P be a u, v-path in $\widehat{G} - \{s, r, x\}$. Since $uv \notin E(\widehat{G})$, P is a V_{t_2}-external u, v-path. Remark 15.4 gives us $t_3 \in N(t_2)$ with $\{u, v\} \subset V_{t_3} \subset \{s, u, v, x\}$. Since P avoids $\{s, r\}$, we have $V_{t_3} \neq V_{t_1}$. Therefore V_{t_3} can only be $\{u, v, x\}$. □

By Lemma 15.2, t_2 has at most 3 neighbors in T. Given this structure of the tree-decomposition (see Figure 18), our aim is now to find a minor from $\mathcal{F}_0'$ by showing the existence of paths between certain vertices. The following lemma helps us do this in certain situations.

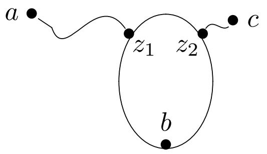


FIG. 19. *The cycle C for Lemma* 16.2.

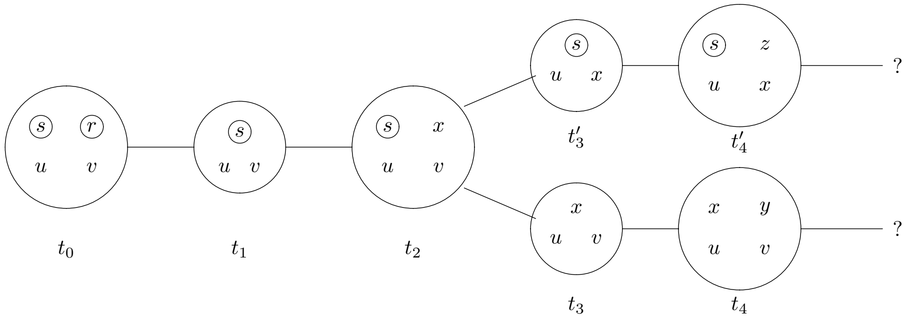


FIG. 20. $(T, \mathcal{V})$ *if* $d_T(t_2) = 3$.

LEMMA 16.2. *Let H be a 2-connected graph with $\{a, b, c\} \subset V(H)$. If $ab \notin E(H)$ and $H - \{a, b\}$ is connected, then $H - a$ contains a cycle C through b, and $H - b$ contains disjoint paths from a and c to $V(C)$ (see Figure 19).*

Proof. There are two internally disjoint paths from $\{a, c\}$ to b in H. Consider the disjoint subpaths from $\{a, c\}$ to $N(b)$ in $H - b$: let P_1 be the a, y_1-path and let P_2 be the c, y_2-path, where $y_1, y_2 \in N(b)$. (P_2 may be trivial.) In $H - \{a, b\}$, there is a path between a vertex z_1 on $P_1 - a$ and a vertex z_2 on P_2. C is formed from this path plus the z_1, y_1-subpath of P_1, the y_2, z_2-subpath of P_2, and b. Thus, we obtain the desired disjoint paths from a to z_1 and from c to z_2. □

We now consider the case when t_2 has degree 3 in T. Denote the third neighbor of t_2 by t'_3. Without loss of generality, we may assume t'_3 has node-set $V_{t'_3} = \{s, u, x\}$.

THEOREM 16.3. *If t_2 has degree 3 in the 3-alternating tree-decomposition of $\widehat{G}$, then $G = F_1$.*

Proof. From Lemma 16.1 and the observation above, we know that the tree-decomposition may be represented as in Figure 20. Since $(T, \mathcal{V})$ is 3-alternating, t_3 and t'_3 are not leaves and have neighbors t_4 and t'_4, respectively, besides t_2. Let y and z denote the distinct new vertices in V_{t_4} and $V_{t'_4}$, respectively.

We will determine the edges or paths in $\widehat{G}$ that are guaranteed by the tree-decomposition. Recall that r is adjacent to u, v, and s and that u and v are nonadjacent. Now, we claim that $sv \in E(\widehat{G})$.

We first prove that there is an s, v-path that avoids $\{u, x, r\}$. Consider the graph G' obtained from $\widehat{G}$ by deleting u, x, and the edge sr. Since $\widehat{G}$ has no sr-diamond-cut, if G' has no s, v-path, then $\{u, x\}$ is a 2-cut in G that separates s from v. Since r is adjacent to v, Lemma 10.5 implies that s is an isolated vertex in $G - \{u, x\}$. However, since $\widehat{G}$ is 3-connected, there is an s, z-path in $\widehat{G} - \{u, x\}$. Such a path must contain the edge sr. This is impossible, since Remark 14.7.1 implies that $\{s, u, x\}$ is a cut-set in $\widehat{G}$ that separates r from z. Hence we may now assume that there is an s, v-path P in $\widehat{G} - \{u, x\} - sr$. If sv is not an edge, then P is V_{t_2}-external, so Remark 15.4 gives us a neighbor of t_2 that contains $\{s, v\}$. Since V_{t_3} and $V_{t'_3}$ do not contain $\{s, v\}$,

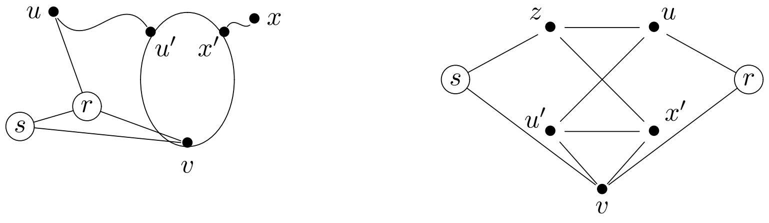


FIG. 21. *Subgraph of $\widehat{G}$ obtained in the proof of Theorem* 16.3, *and* F_1.

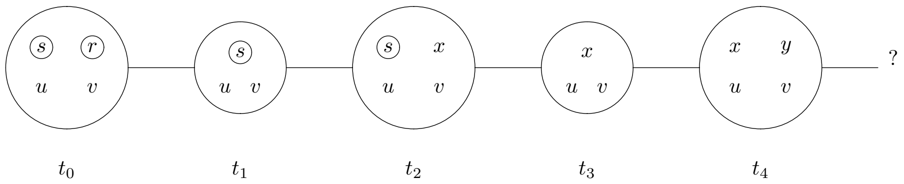


FIG. 22. $(T, \mathcal{V})$ *if* $d_T(t_2) = 2$.

we conclude that P must visit precisely s, r, v, which contradicts the choice of P. Therefore sv must be an edge in $\widehat{G}$.

We will now apply Lemma 16.2 with $H = \widehat{G}^-$ and $(a, b, c) = (u, v, x)$ to obtain the paths needed to find F_1 as a minor. Certainly, H is 2-connected. Furthermore, since $N_{\widehat{G}}(r) = \{u, v, s\}$, it follows that if $\{u, v, s, r\}$ is a cut-set in $\widehat{G}$, then $\{u, v, s\}$ is a cut-set that separates $\widehat{G}$ into at least 3 components. This contradicts Theorem 13.3, so $\{u, v\}$ is not a cut-set in $\widehat{G}^-$. Thus Lemma 16.2 gives a cycle in $\widehat{G}^- - u$ through v and disjoint paths in $\widehat{G}^- - v$ from u and x to some u' and x' on the cycle (see Figure 21). Since $\{s, u, x\}$ is a cut-set in $\widehat{G}$ that separates v from z (by Remark 14.7.1), the paths and cycle cannot intersect the component of $\widehat{G} - V_{t'_3}$ that contains z.

By the 3-connectedness of $\widehat{G}$ we can also find three internally disjoint paths from z to $\{s, u, x\}$, with all internal vertices contained in the component of $\widehat{G} - \{s, u, x\}$ that contains z. We obtain a subdivision of F_1^+ with branch vertices s, r, z, u, u', x', and v, establishing the theorem. ▯

Thus, we may assume that in the tree-decomposition of $\widehat{G}$, t_2 has only two neighbors t_1 and t_3 whose corresponding node-sets are $\{s, u, v\}$ and $\{u, v, x\}$. Since $(T, \mathcal{V})$ is 3-alternating, t_3 is not a leaf. Let t_4 denote the other neighbor of t_3 and let V_{t_4} be $\{u, v, x, y\}$. Thus $N_{\widehat{G}}(s) \subseteq \{r, u, v, x\}$ by Definition 14.1.2. Since $\widehat{G}$ is 3-connected, this forces sx to be an edge of $\widehat{G}$, since otherwise $\{u, v\}$ would be a 2-cut in $\widehat{G}$ separating s and r from x. Also s has at least one neighbor besides r and x, so without loss of generality we may assume that $N_{\widehat{G}}(s) \supseteq \{r, u, x\}$.

If v is also a neighbor of s, then the next lemma asserts that we find T_s as an sr-minor.

LEMMA 16.4. *If the structure of the* 3*-alternating tree-decomposition of* $\widehat{G}$ *is as shown in Figure* 22 *and* $sv \in E(\widehat{G})$, *then* $G = T_s$.

Proof. Since $N_{\widehat{G}}(s) = \{r, u, v, x\}$ and $N_{\widehat{G}}(r) = \{s, u, v\}$, to find T_s as an sr-minor it suffices to find appropriate paths in $\widehat{G}^-$ from $\{u, v, x\}$ to some other vertex z. Since $\widehat{G}^-$ is 2-connected, there are two internally disjoint paths P_1, P_2 from u to v in $\widehat{G}^-$. Also, since $\{u, v\}$ is not a cut-set in $\widehat{G}$, $\widehat{G} - \{u, v\}$ contains a (possibly trivial) path

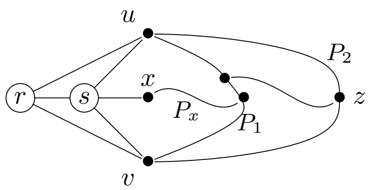


FIG. 23. *Subgraph of $\widehat{G}$ in proof of Lemma* 16.4.

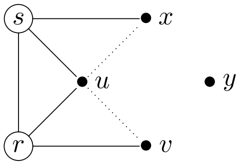


FIG. 24. *Known subgraph of $\widehat{G}$ thus far.*

from x to $P_1 \cup P_2 - \{u, v\}$. We may assume that a minimal such path P_x goes from x to P_1 in $\widehat{G} - P_2$. Since $\widehat{G}$ is 3-connected, there is a path in $\widehat{G} - \{u, v\}$ from some vertex on $P_1 \cup P_x$ to some vertex z on P_2, and it clearly cannot contain s or r. Consider a minimal such path and contract P_x; this yields T_s^+ as an sr-minor of $\widehat{G}$. □

17. Hunting the last minor.

The final theorem asserts that in the remaining case we can find F_1 as an sr-minor of G, regardless of the degree of t_4 and the choice of node-sets corresponding to neighbors of t_4.

THEOREM 17.1. *If the structure of the* 3*-alternating tree-decomposition of $\widehat{G}$ is as shown in Figure* 22 *and $sv \notin E(\widehat{G})$, then $G = F_1$.*

Proof. According to the remarks preceding Lemma 16.4, we may now assume that $N_{\widehat{G}}(r) = \{s, u, v\}$ and $N_{\widehat{G}}(s) = \{r, u, x\}$. We also showed that $uv \notin E(\widehat{G})$, and now we can similarly show that $ux \notin E(\widehat{G})$: Since $N_{\widehat{G}}(s) = \{r, u, x\}$, $\{u, x\}$ is a 2-cut in G, so Lemma 10.6 implies that ux is not an edge of the original graph G. Also, $\{u, x\}$ is not a diamond-cut in G^+, since otherwise by Lemma 10.3 the diamond component would have to intersect G_π, and applying Lemma 10.2 would contradict Theorem 11.5. Thus $ux \notin E(\widehat{G})$ (see Figure 24).

Since u is not adjacent to x or v, but $\widehat{G}^-$ is 2-connected, u has degree at least two and hence has a neighbor other than y (y need not be adjacent to u.) Thus there must be a node in the tree whose corresponding node-set contains both u and this neighbor. Hence t_4 has a neighbor t_5 (other than t_3) such that $u \in V_{t_5}$.

By Lemma 15.2, t_4 has degree at most 3, so t_4 has one or two neighbors other than t_3. Since the node-sets corresponding to these neighbors are 3-subsets of $\{u, v, x, y\}$, they can only be $\{u, v, y\}$, $\{u, x, y\}$, or $\{v, x, y\}$. The proof now breaks into three cases.

Case 1. t_4 has degree 3 and the node-sets corresponding to its neighbors t_5 and t'_5 are $\{u, x, y\}$ and $\{u, v, y\}$ respectively, as shown in Figure 25.

Let T_5 be the component of $T - t_4$ that contains t_5. By Remark 14.7.1, we may let R_5 be a component of $\widehat{G} - V_{t_4}$ whose vertices are contained in the node-sets corresponding to nodes of T_5 and no other node-sets. Similarly define T'_5 and R'_5 for t'_5.

We claim that $xv \in E(\widehat{G})$. We first prove that there is an x, v-path P that avoids $\{s, r, u, y\}$. Consider the graph G' obtained by deleting u, y and the edge sr from $\widehat{G}$,

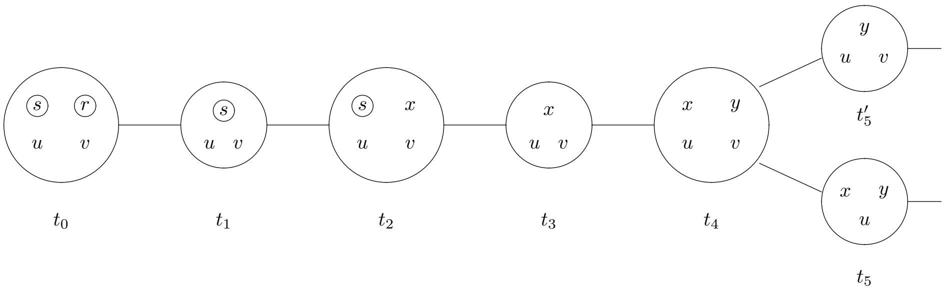


FIG. 25. *Structure of* $(T,\mathcal{V})$ *in Case* 1.

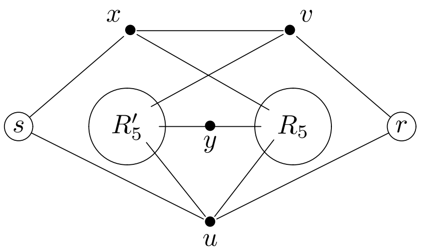


FIG. 26. *A subgraph of* $\widehat{G}$ *from Case* 1 *of the Proof of Theorem* 17.1.

and suppose that there is no x,v-path in G'. Thus, Remark 12.4 implies that $\{u,y\}$ is a 2-cut in G that separates x from v. However, the component with x contains s and the component with v contains r, which contradicts Lemma 10.5. Since s and r each have degree 1 in G', we thus have an x,v-path P in $\widehat{G}$ avoiding $\{s,r,u,y\}$. Observe that $\{x,v\} \not\subseteq V_{t_5}$, $\{x,v\} \not\subseteq V_{t_5'}$, and P doesn't intersect $\{s,r\}$. Hence Remark 15.4 implies that P is not a V_{t_4}-external path. Since P avoids $\{u,y\}$, P must be the edge xv.

Consider the components R_5 and R_5' of $\widehat{G} - V_{t_4}$. By 3-connectedness, there are edges from R_5 to each vertex of V_{t_5} and edges from R_5' to each vertex of $V_{t_5'}$ as in Figure 26. Contracting y into R_5' and contracting R_5 and R_5' yields F_1^+ as an *sr*-minor of $\widehat{G}$, which suffices.

Case 2. $V_{t_5} = \{u,x,y\}$, and no neighbor of t_4 has $\{u,v,y\}$ as a node-set.

In this case, $\{v,x,y\}$ may or may not be a node-set. Remark 15.4 applied to V_{t_4} asserts that every internal vertex of a V_{t_4}-external u,v-path must be in $\{s,r\}$. Since uv is not an edge in $\widehat{G}$, $\{x,y\}$ is a 2-cut in $\widehat{G}^-$ separating u from v. By 2-connectedness $\widehat{G}^-$ contains two internally disjoint paths $P_{v,x}$ and $P_{v,y}$ from v to x and y, respectively (see Figure 27(a)). Since s is isolated in $\widehat{G}-\{u,x,r\}$, Theorem 13.3 implies that $\widehat{G}-\{u,x,r,s\}$ is connected. Since $\widehat{G}^- -\{u,x\}$ is connected, we can apply Lemma 16.2 to $H = \widehat{G}^-$ with $(a,b,c) = (x,u,y)$. This gives a cycle C through u in $\widehat{G}^- - x$ and disjoint paths $P_{x,C}$ and $P_{y,C}$ in $\widehat{G}^- - u$ from x and y to the cycle (see Figure 27(b)).

Observe that $P_{v,x} \cup P_{v,y}$ doesn't intersect $C \cup P_{x,C} \cup P_{y,C} - \{x,y\}$, since $\{x,y\}$ separates u from v in $\widehat{G}^-$. Thus we obtain F_1 as an *sr*-minor (see Figure 27(c)).

Case 3. $V_{t_5} = \{u,v,y\}$, and no neighbor of t_4 has $\{u,x,y\}$ as a node-set.

The proof is similar to that of Case 2, but we simply interchange s with r and x with v. ◻

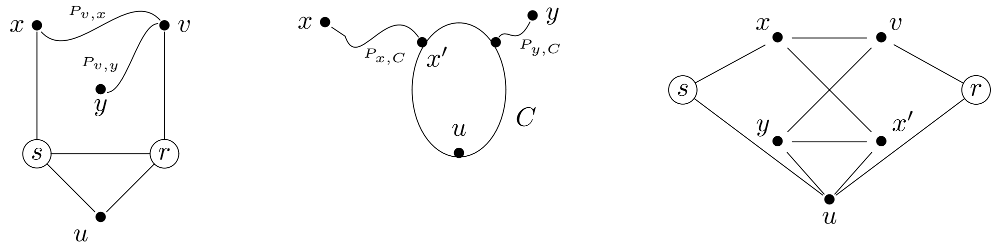


FIG. 27. *Case 2: two subgraphs of $\widehat{G}$ that share only $\{u, x, y\}$ and the resulting F_1-minor.*

We have now exhausted all possibilities for the tree-decomposition of an obstruction. Thus we have shown that every graph from the family of obstructions to headerless reliable single message transmission contains some graph from $\mathcal{F}_0'$ as an *sr*-minor. As discussed in section 8, this completes the proof of Theorem 8.1.

REFERENCES

[1] M. Adler and F. E. Fich, *The complexity of end-to-end communication in memoryless networks*, in: 18th Annual ACM Symposium on Principles of Distributed Computing, May 1999, pp. 239–248.

[2] M. Adler, F. E. Fich, L. A. Goldberg, and M. Paterson, *Tight size bounds for packet headers in narrow meshes*, in Proceedings of the 27th International Colloquium on Automata, Languages and Programming (ICALP), July 2000.

[3] S. Arnborg, D. G. Corneil, and A. Proskurowski, *Forbidden minors characterization of partial 3-trees*, Discrete Math., 80 (1990), pp. 1–19.

[4] S. Arnborg, J. Lagergren, and D. Seese, *Easy problems for tree-decomposable graphs*, J. Algorithms, 12 (1991), pp. 308–340.

[5] H. L. Bodlaender, *A partial k-arboretum of graphs with bounded treewidth*, Theoret. Comput. Sci., 209 (1998), pp. 1–45.

[6] H. L. Bodlaender, *A linear time algorithm for finding tree-decompositions of small treewidth*, SIAM J. Comput., 25 (1996), pp. 1305–1317.

[7] R. Diestel, *Graph Theory*, Graduate Texts in Mathematics 173, Springer-Verlag, New York, 2000.

[8] R. J. Duffin, *Topology of series-parallel networks*, J. Math. Anal. Appl., 10 (1965), pp. 303–318.

[9] F. E. Fich, *End-to-end communication*, in Proceedings of the 2nd International Conference on Principles of Distributed Systems, Amiens, France, 1998, pp. 37–43.

[10] P. Fraigniaud and C. Gavoille, *Lower bounds for oblivious single-message end-to-end communication*, in Proceedings of the 17th International Symposium on Distributed Computing (DISC) 2003.

[11] J. Matoušek and R. Thomas, *Algorithms finding tree-decompositions of graphs*, J. Algorithms, 12 (1991), pp. 1–22.

[12] M. J. Pelsmajer, *Equitable List Coloring, Induced Linear Forests, and Routing in Rooted Graphs*, Ph.D. thesis, Department of Mathematics, University of Illinois, Urbana, 2002.

[13] J. Postel, *Internet Protocol*, Network Working Group Request for Comments 791, Sept. 1981.

[14] N. Robertson and P. D. Seymour, *Graph minors* V. *Excluding a planar graph*, J. Combin. Theory Ser. B, 41 (1986), pp. 92–114.

[15] N. Robertson and P. D. Seymour, *Graph minors* XX. *Wagner's conjecture*, J. Combin. Theory Ser. B, 92 (2004), pp. 325–357.

[16] N. Robertson, P. D. Seymour, and R. Thomas, *Hadwiger's conjecture for K_6-free graphs*, Combinatorica, 13 (1993), pp. 279–361.

[17] N. Robertson, P. D. Seymour, and R. Thomas, *Quickly excluding a planar graph*, J. Combin. Theory Ser. B, 62 (1994), pp. 323–348.

[18] R. Thomas, personal communication.

[19] D. B. West, *Introduction to Graph Theory*, 2nd ed., Prentice-Hall, Upper Saddle River, NJ, 2001.

SIAM J. DISCRETE MATH.
Vol. 19, No. 4, pp. 848–880

CHAIN DECOMPOSITIONS OF 4-CONNECTED GRAPHS[*]

SEAN CURRAN[†], ORLANDO LEE[‡], AND XINGXING YU[§]

Abstract. In this paper we give a decomposition of a 4-connected graph G into nonseparating chains, which is similar to an ear decomposition of a 2-connected graph. We also give an $O(|V(G)|^2|E(G)|)$ algorithm that constructs such a decomposition. In applications, the asymptotic performance can often be improved to $O(|V(G)|^3)$. This decomposition will be used to find four independent spanning trees in a 4-connected graph.

Key words. connectivity, nonseparating, good chain, chain decomposition, algorithm

AMS subject classifications. 05C40, 05C85, 05C38, 05C75

DOI. 10.1137/S0895480103434592

1. Introduction.

In [1], Cheriyan and Maheshwari gave an $O(|V(G)|^2)$ algorithm for finding a "nonseparating ear decomposition" of a 3-connected graph G, and they used this decomposition to construct three independent spanning trees in a 3-connected graph.

In this paper we give a 4-connected version of the nonseparating ear decomposition of Cheriyan and Maheshwari and an $O(|V(G)|^2|E(G)|)$ algorithm for finding such a decomposition. This will be used in a forthcoming paper to find four independent spanning trees in an arbitrary 4-connected graph G, where the asymptotic performance can be improved to $O(|V(G)|^3)$.

We use the definitions and notation given in [2]. Some of those definitions are quite long, so we simply refer the readers to [2]. In particular, see [2] for definitions of *chain* (Definition 1.3 of [2]), *planar chain* (Definition 1.4 of [2]), *cyclic chain* (Definition 4.2 of [2]), and *planar cyclic chain* (Definition 4.3 of [2]). Intuitively, the roles of planar chains and planar cyclic chains in our decompsitions of 4-connected graphs are similar to those of paths and cycles in ear decompositions of 2-connected graphs.

In [2], we showed how to find the first planar chain in our decomposition of 4-connected graphs. The other chains in our decomposition can be classified into four types, as described below. The first three types are planar chains as defined in Definition 1.1. The fourth type is not a planar chain (but almost planar as we will see), and it is defined in Definition 1.2. See Figure 1 for illustrations of Definitions 1.1 and 1.2.

DEFINITION 1.1. *Let G be a graph, let F be a subgraph of G, and let $r \in V(F)$. Let H be a planar x-y chain in G such that $V(H) - \{x, y\} \subseteq V(G) - V(F)$.*

[*] Received by the editors September 17, 2003; accepted for publication (in revised form) June 3, 2005; published electronically December 30, 2005.
http://www.siam.org/journals/sidma/19-4/43459.html

[†] School of Mathematics, Georgia Institute of Technology, Atlanta, Georgia 30332 (curran@math.gatech.edu). The work of this author was partially supported by an NSF VIGRE grant.

[‡] Instituto de Computação, Universidade Estadual de Campinas, Avenida Albert Einstein, 1251, Caixa Postal 6176, 13083-970 Campinas, SP, Brazil (lee@ic.unicamp.br). The work of this author was supported by CNPq (Proc: 200611/00-3), Brazil.

[§] School of Mathematics, Georgia Institute of Technology, Atlanta, Georgia 30332 and Center for Combinatorics, LPMC, Nankai University, Tianjin, China 300071 (yu@math.gatech.edu). The work of this author was partially supported by NSF grants DMS9970527 and DMS0245530 and by NSA grant MDA904-03-1-0052.

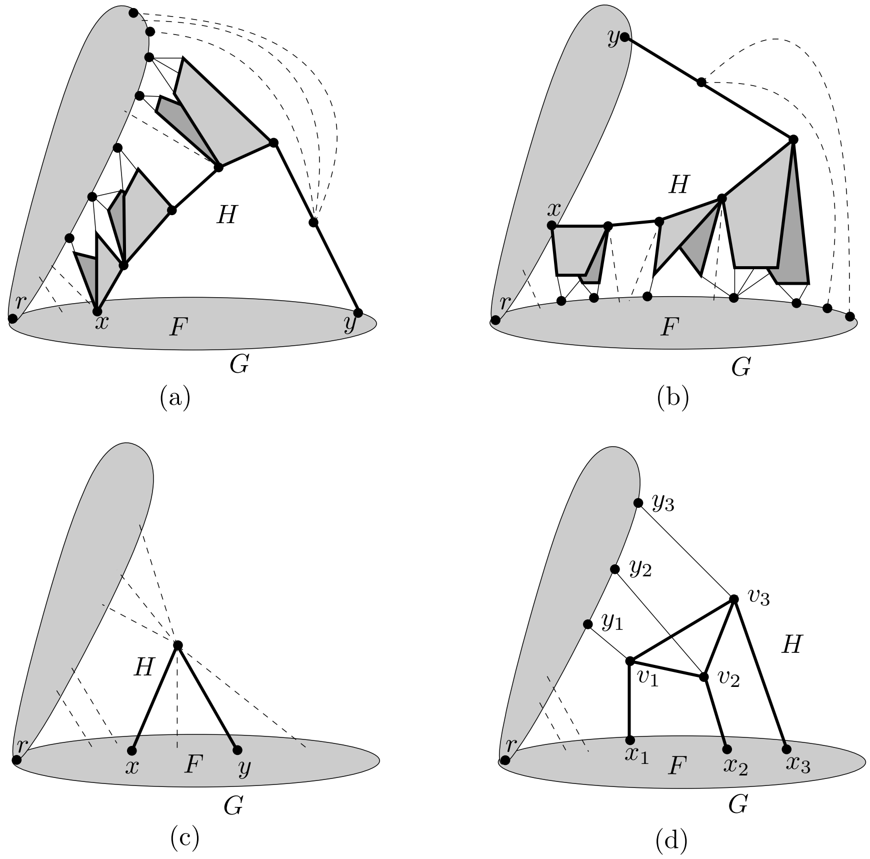


FIG. 1. (a) *An up* F*-chain,* (b) *a down* F*-chain,* (c) *an elementary* F*-chain, and* (d) *a triangle* F*-chain. The dashed edges need not exist.*

We say that

(a) *H is an up F-chain if $\{x, y\} \subseteq V(F)$ and $N_G(H - \{x, y\}) \subseteq (V(G) - V(F - r)) \cup \{x, y\}$;*

(b) *H is a down F-chain if $\{x, y\} \subseteq V(G) - V(F - r)$ and $N_G(H - \{x, y\}) \subseteq V(F - r) \cup \{x, y\}$; and*

(c) *H is an elementary F-chain if $\{x, y\} \subseteq V(F)$ and H is an x-y path of length two.*

In any of the three cases above we say that H is a planar *x-y F-chain in G (or simply, a* planar F-chain*). For an x-y chain H we let $I(H) := V(H) - \{x, y\}$, and for a cyclic chain H we let $I(H) := V(H)$.*

For a graph G, a subgraph H of G, and $S \subseteq V(G) \cup E(G)$, we let $H + S$ denote the graph with vertex set $V(H) \cup (S \cap V(G))$ and edge set $E(H) \cup (S \cap E(G))$.

DEFINITION 1.2. *Let G be a graph, let F be a subgraph of G, and let $r \in V(F)$. Suppose that $\{v_1, v_2, v_3\} \subseteq V(G) - V(F)$ induces a triangle T in G, and for each $1 \le i \le 3$, v_i has exactly one neighbor x_i in $V(F - r)$ and exactly one neighbor y_i in $V(G) - (V(F) \cup V(T))$ (thus, each v_i has degree four in G). Moreover, assume that x_1, x_2, x_3 are distinct and y_1, y_2, y_3 are distinct. Then we say that $H := T + \{x_1, x_2, x_3, v_1x_1, v_2x_2, v_3x_3\}$ is a* triangle F-chain *in G. We let $I(H) := \{v_1, v_2, v_3\}$.*

The definitions above depend on the choice of r and F, but in spite of this, whenever we use these concepts in this paper, it should be clear which pair r, F we refer to.

Definition 1.3. *Let G be a graph, let F be a subgraph of G, and let $r \in V(F)$. By a* good F-chain *in G, we mean an up F-chain, a down F-chain, an elementary F-chain, or a triangle F-chain.*

We are now ready to describe a chain decomposition, which is similar to an ear decomposition.

Definition 1.4. *Let G be a graph, let $r \in V(G)$, and let $H_1, \dots, H_t$ be chains in G, where $t \geq 2$. We say that $(H_1, \dots, H_t)$ is a* nonseparating chain decomposition *of G* rooted at r *if the following conditions hold:*

(i) *H_1 is a planar cyclic chain in G rooted at r;*

(ii) *for each $i = 2, \dots, t-1$, H_i is a good $G[\bigcup_{j=1}^{i-1} I(H_j)]$-chain in G;*

(iii) *$H_t = G - (\bigcup_{j=1}^{t-1} I(H_j) - \{r\})$ is a planar cyclic chain in G rooted at r; and*

(iv) *for each $i = 1, \dots, t-1$, both $G[\bigcup_{j=1}^{i} I(H_j)]$ and $G - (\bigcup_{j=1}^{i} I(H_j) - \{r\})$ are 2-connected.*

The chains $H_2, \dots, H_{t-1}$ are called internal chains *of the nonseparating chain decomposition. If ra is a piece of H_1, then we say that $H_1, \dots, H_t$ is a nonseparating chain decomposition of G* starting at ra.

The main result of this paper is the following.

Theorem 1.5. *Let G be a 4-connected graph, let $r \in V(G)$, and let $ra \in E(G)$. Then G has a nonseparating chain decomposition rooted at r starting at ra, and such a decomposition can be found in $O(|V(G)|^2|E(G)|)$ time.*

The existence of the first chain H_1 of the chain decomposition is guaranteed by the next result which corresponds to Theorem 4.4 of [2].

Theorem 1.6. *Let G be a 4-connected graph, and let $ra \in E(G)$. Then there exists a planar cyclic chain H in G rooted at r such that ra is a piece of H and $G - (V(H) - \{r\})$ is 2-connected. Moreover, such a chain can be found in $O(|V(G)||E(G)|)$ time.*

In order to construct the internal chains of the chain decomposition in Theorem 1.5, we need the following result which is Theorem 1.6 of [2].

Theorem 1.7. *Let G be a graph, let $\{a, b\} \subseteq V(G)$, and let P be a nonseparating induced a-b path in G. Let B_P be a nontrivial block of $G - V(P)$, and let $X_P := N_G(G - V(B_P))$. Suppose $G - (V(B_P) - X_P)$ is $(4, X_P \cup \{a, b\})$-connected. Then there exists a planar a-b chain H in G such that $G - V(H)$ is 2-connected and $B_P \subseteq G - V(H)$. Moreover, such a chain can be found in $O(|V(G)||E(G)|)$ time.*

The rest of this paper is organized as follows. In section 2 we recall some lemmas proved in [2] and provide some new auxiliary lemmas concerning nonseparating induced paths. In section 3 we prove a technical result, which will be used to find the internal chains of a nonseparating chain decomposition. Finally, in section 4 we complete the proof of Theorem 1.5.

2. Nonseparating paths.

In this section we state and prove some results concerning nonseparating induced paths which will be used later. First, we state two lemmas without proof, which are Lemmas 2.3 and 2.4 of [2], respectively.

Lemma 2.1. *Let G be a connected graph, $S \subseteq V(G)$, $\{a, a'\} \subseteq S$, and let P be an a-a' path in G. Suppose*

(i) *G is $(3, S)$-connected, and*

(ii) *$S - \{a, a'\}$ is contained in a component U of $G - V(P)$.*

Then there exists a nonseparating induced a-a' path P' in G such that $V(P') \cap V(U) = \emptyset$. Moreover, such a path can be found in $O(|V(G)| + |E(G)|)$ time.

LEMMA 2.2. *Let G be a graph and $S := \{a, a', b, b'\} \subseteq V(G)$. Suppose that G is $(4, S)$-connected. Then exactly one of the following holds:*

(1) *there exists a nonseparating induced a-a' path P' in G such that $V(P') \cap \{b, b'\} = \emptyset$;*

(2) *(G, a, b, a', b') is planar.*

Moreover, one can in $O(|V(G)| + |E(G)|)$ time find a path as in (i) *or certify that* (ii) *holds.*

Note our use of "prime" notation in the statements of the lemmas. The reader should not infer that the paths labeled P' are derived from an assumed path P. We reserve P to denote a particular path specified in section 3, and we therefore label paths P' in the statements of our lemmas. We hope this will sidestep any source of confusion when these lemmas are applied.

The next lemma is a variation of Lemma 2.1 (and Lemma 2.2 as well) in which we prove the existence of a specific nonseparating induced path. However, here it is not possible to specify the ends of the desired path. Moreover, in the hypotheses of Lemma 2.3 there are some technical conditions which arise when we try to produce an internal chain. Note that conditions (iii), (iv), and (v) of Lemma 2.3 are automatically satisfied if G is $(4, S \cup \{b, b'\})$-connected. Actually, this is the case in all applications of Lemma 2.3 with the exception of the proof of Lemma 3.15, where the more complicated conditions are required.

LEMMA 2.3. *Let G be a graph, let $S \subset V(G)$, and let $\{b, b'\} \subseteq V(G) - S$. Suppose*

(i) *$G - S$ is 2-connected,*

(ii) *every element of S has a neighbor in $V(G) - (S \cup \{b, b'\})$,*

(iii) *G is $(3, S \cup \{b, b'\})$-connected,*

(iv) *if $|S| = 2$, then G is $(4, S \cup \{b, b'\})$-connected, and*

(v) *if $|S| \geq 3$, then there exists some component of $G - (S \cup \{b, b'\})$ which has at least two neighbors in S.*

Then exactly one of the following holds:

(1) *there exist $a, a' \in S$ and an induced a-a' path P' in G such that $V(P') \cap \{b, b'\} = \emptyset$, $V(P') \cap S = \{a, a'\}$, and $G - (V(P') \cup S)$ is connected;*

(2) *$|S| = 2$, and the elements of S can be labeled as a, a' such that (G, a, b, a', b') is planar.*

Moreover, one can in $O(|V(G)| + |E(G)|)$ time find a path as in (1) *or certify that* (2) *holds.*

Proof. First, suppose that $|S| = 2$. Let a, a' denote the vertices in S. By (iv) G is $(4, \{a, a', b, b'\})$-connected. Thus, by Lemma 2.2 exactly one of the following holds:

(a) there exists a nonseparating induced a-a' path P' such that $V(P') \cap \{b, b'\} = \emptyset$; or

(b) (G, a, b, a', b') is planar.

Moreover, one can in $O(|V(G)| + |E(G)|)$ time find a path as in (a) or certify that (b) holds. If (a) holds, then P' is the required path in (1). If (b) holds, then (2) holds.

Thus, we may assume that $|S| \geq 3$. First, we prove the following.

Claim. There exist $a, a^* \in S$ and an a-a^* path Q in $G - (S - \{a, a^*\})$ such that b and b' are contained in a component of $G - V(Q)$. Moreover, such a path can be found in $O(|V(G)| + |E(G)|)$ time.

Proof of Claim. We consider two cases. See Figure 2 for an illustration of the outcomes of Lemma 2.3.

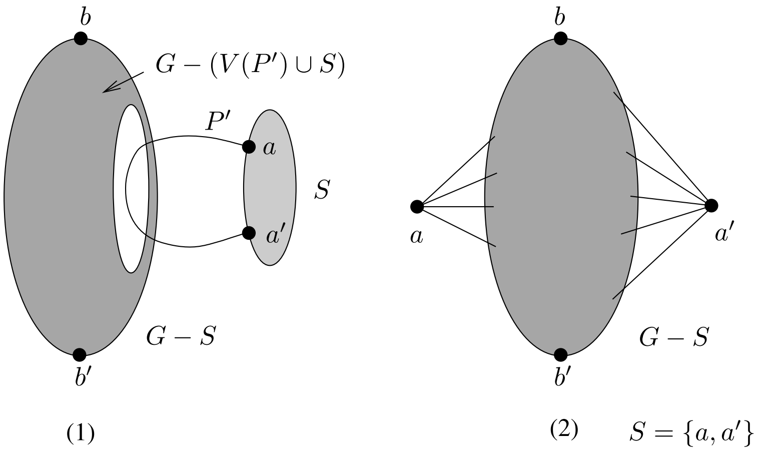


FIG. 2. *Outcomes in Lemma* 2.3.

Case 1. $G-(S\cup\{b,b'\})$ is not connected.

In this case, there exist edge-disjoint subgraphs G_1, G_2 of $G-S$ such that $G_1\cup G_2 = G-S$, $V(G_1)\cap V(G_2)=\{b,b'\}$, and $|V(G_1)|\geq 3\leq |V(G_2)|$. Note that such a partition can be found in $O(|V(G)|+|E(G)|)$ time. Since $G-S$ is 2-connected (by (i)), both G_1 and G_2 are connected.

By (v) there exists some component K of $G-(S\cup\{b,b'\})$ which has at least two neighbors in S. Note that such a component can be found in $O(|V(G)|+|E(G)|)$ time. We may assume that $V(K)\subseteq V(G_1)$. Let $a,a^*\in N_G(K)\cap S$, and let Q be an a-a^* path in $G[V(K)\cup\{a,a^*\}]$. Since G_2 is connected, b,b' are contained in a component of $G-V(Q)$. Moreover, such a path can be found in $O(|V(G)|+|E(G)|)$ time.

Case 2. $G-(S\cup\{b,b'\})$ is connected.

Since $G-S$ is 2-connected by (i), one can find in $O(|V(G)|+|E(G)|)$ time two internally disjoint b-b' paths P_1,P_2 in $G-S$. Let a_1,a_2,a_3 be distinct vertices in S. For $i=1,2,3$, let $v_i\in N_G(a_i)\subseteq V(G)-(S\cup\{b,b'\})$ (they exist by (ii)). Since $G-(S\cup\{b,b'\})$ is connected by assumption, for each $i=1,2,3$, there exists a path Q_i from v_i to some vertex u_i in $(V(P_1)\cup V(P_2))-\{b,b'\}$ internally disjoint from $V(P_1)\cup V(P_2)$. Moreover, such paths can be found in $O(|V(G)|+|E(G)|)$ time. Note that at least two (not necessarily distinct) vertices in u_1,u_2,u_3 lie on the same path $P_1-\{b,b'\}$ or $P_2-\{b,b'\}$. By symmetry, we may assume that $u_1,u_2\in V(P_1)-\{b,b'\}$. Then there exist disjoint paths in $G-(S-\{a_1,a_2\})$ from a_1 to a_2 (the path contained in $Q_1\cup Q_2\cup(P_1-\{b,b'\})$) and from b to b' (the path P_2), respectively. Thus, the result follows by taking $a=a_1$ and $a^*=a_2$. Moreover, it is not hard to see that such paths can be found in $O(|V(G)|+|E(G)|)$ time. ▯

Now given a,a^* and Q, we will describe how to find $a'\in S$ and an induced a-a' path P' such that $V(P')\cap\{b,b'\}=\emptyset$, $V(P')\cap S=\{a,a'\}$, and $G-(V(P')\cup S)$ is connected. Let G' be the graph obtained from G by identifying the vertices in $S-\{a\}$ to a single vertex a'' and removing the resulting multiple edges. Let $S':=\{a,a'',b,b'\}$.

We claim that G' is $(3,S')$-connected. Suppose for a contradiction that there exists $T\subseteq V(G')$ such that $|T|\leq 2$ and $G'-T$ has a component K with $V(K)\cap S'=\emptyset$. Clearly $a''\in T$ because G is $(3,S\cup\{b,b'\})$-connected (by (iii)); then either $a\in T$ or

$T - \{a''\}$ is a vertex cut of $G - S$, which is a contradiction since $G - S$ is 2-connected (by (i)). Thus, G' is $(3, S')$-connected.

Note that the a-a^* path Q in G corresponds to an a-a'' path P in G', and $S' - \{a, a''\} = \{b, b'\}$ is contained in a component U of $G' - V(P)$. Thus, the hypotheses of Lemma 2.1 are satisfied with G', S', P, a, a'', U as G, S, P, a, a', U, respectively. Hence, there exists a nonseparating induced a-a'' path P'' in G' such that $V(P'') \cap V(U) = \emptyset$. Moreover, such a path P'' can be found in $O(|V(G')| + |E(G')|)$ time (hence, in $O(|V(G)| + |E(G)|)$ time). The path P'' corresponds to an induced a-a' path P' in G for some $a' \in S - \{a\}$ such that $V(P') \cap \{b, b'\} = \emptyset$ and $V(P') \cap S = \{a, a'\}$. Since P'' is nonseparating in G', $G - (V(P') \cup S)$ is connected. Therefore, a, a' and P' satisfy (1), and they can be found in $O(|V(G)| + |E(G)|)$ time. ◻

The following lemma is a variation of Lemma 2.3 (by letting $b = b'$), and its proof is essentially the same. For the sake of completeness, we include it here.

LEMMA 2.4. *Let G be a graph, let $S \subseteq V(G)$, and let $b \in V(G) - S$. Suppose*

(i) *$G - S$ is 2-connected,*
(ii) *every element of S has a neighbor in $V(G) - (S \cup \{b\})$, and*
(iii) *G is $(3, S \cup \{b\})$-connected.*

Then there exist $a, a' \in S$ and an induced a-a' path P' in G such that $V(P') \cap \{b\} = \emptyset$, $V(P') \cap S = \{a, a'\}$, and $G - (V(P') \cup S)$ is connected. Moreover, such a path can be found in $O(|V(G)| + |E(G)|)$ time.

Proof. Since G is $(3, S \cup \{b\})$-connected (by (iii)), $|S| \geq 2$, so let $a, a^* \in S$. Since $G - S$ is 2-connected (by (i)), $G - (S \cup \{b\})$ is connected. Since a and a^* have a neighbor in $V(G) - (S \cup \{b\})$ (by (ii)), there exists an a-a^* path Q in $G - ((S - \{a, a^*\}) \cup \{b\})$. Clearly, such a path can be found in $O(|V(G)| + |E(G)|)$ time.

Let G' be the graph obtained from G by identifying the vertices in $S - \{a\}$ to a single vertex a'' and removing the resulting multiple edges. Let $S' := \{a, a'', b\}$.

We claim that G' is $(3, S')$-connected. Suppose for a contradiction that there exists $T \subseteq V(G')$ such that $|T| \leq 2$ and $G' - T$ has a component K with $V(K) \cap S' = \emptyset$. Clearly, $a'' \in T$ because G is $(3, S \cup \{b\})$-connected (by (iii)). But then either $a \in T$ or $T - \{a''\}$ is a vertex cut of $G - S$, which is a contradiction since $G - S$ is 2-connected (by (i)). Thus, G' is $(3, S')$-connected.

Note that the a-a^* path Q in G corresponds to an a-a'' path P in G', and $S' - \{a, a''\} = \{b\}$ is contained in a component U of $G' - V(P)$. Thus, by Lemma 2.1 (with G', S', P, a, a'', U as G, S, P, a, a', U, respectively), there exists a nonseparating induced a-a'' path P'' in G' such that $V(P'') \cap V(U) = \emptyset$. Moreover, such a path P'' can be found in $O(|V(G')| + |E(G')|)$ time (and hence, in $O(|V(G)| + |E(G)|)$ time). The path P'' corresponds to an induced a-a' path P' in G for some $a' \in S - \{a\}$ such that $V(P') \cap \{b\} = \emptyset$ and $V(P') \cap S = \{a, a'\}$. Since P'' is nonseparating in G', $G - (V(P') \cup S)$ is connected. So a, a' and P' are as required, and they can be found in $O(|V(G)| + |E(G)|)$ time. ◻

Some results and algorithms which we use here require that we find an embedding of a planar graph (G, a, b, c, d) in a closed disk such that a, b, c, d occur on the boundary of the disk in that cyclic order. This can be done in linear time using an algorithm of Hopcroft and Tarjan [4] (or a more recent algorithm by Hsu and Shih [5]). For convenience, we state this result as our next lemma.

LEMMA 2.5. *Let (G, a, b, c, d) be a planar graph. Then one can find in $O(|V(G)| + |E(G)|)$ time an embedding of G in a closed disk such that a, b, c, d occur on the boundary of the disk in that cyclic order.*

Let (G, a, b, a', b') be a planar graph. Then any a-a' path in $G - \{b, b'\}$ separates b from b'. The next lemma shows that one can find efficiently an a-a' path P' in $G - \{b, b'\}$ such that $G - V(P')$ has exactly two components. This will be used in section 3.

LEMMA 2.6. *Let (G, a, b, a', b') be a planar graph with $|V(G)| \geq 5$ and suppose G is $(4, \{a, a', b, b'\})$-connected. Then there exists an induced a-a' path P' in G such that $G - V(P')$ has exactly two components K and K' with $b \in V(K)$ and $b' \in V(K')$. Moreover, such a path can be found in $O(|V(G)| + |E(G)|)$ time.*

Proof. Take an embedding of G in a closed disk such that a, b, a', b' occur on the boundary of the disk in the cyclic order listed. By Lemma 2.5, this can be done in $O(|V(G)| + |E(G)|)$ time. Let $G' := (G - b') + \{ab, a'b\}$.

We claim that G' is 2-connected. Suppose for a contradiction that G' is not 2-connected. Let x be a cut vertex of G'. Since $|V(G)| \geq 5$ and G is $(4, \{a, a', b, b'\})$-connected, $G - \{b, b'\}$ contains an a-a' path, and hence, $\{a, a', b\}$ is contained in a cycle in G'. Therefore, $\{a, a', b\}$ is contained in an x-bridge of G', and G' has another x-bridge B such that $(V(B) - \{x\}) \cap \{a, a', b\} = \emptyset$. Hence, $B - x$ is a component of $G - T$, where $T := \{x, b'\}$ and $(V(B) - \{x\}) \cap \{a, a', b, b'\} = \emptyset$, which contradicts the assumption that G is $(4, \{a, a', b, b'\})$-connected.

Thus, we can assume that $ab, a'b$ are in the cycle bounding the infinite face of G'. Let P' be the a-a' subpath of this cycle which avoids b. Note that $N_G(b') \subseteq V(P')$ and P' can be computed in $O(|V(G)| + |E(G)|)$ time.

We claim that $G' - V(P')$ is connected. Suppose for a contradiction that $G' - V(P')$ is not connected. Let $\mathcal{K}$ be the set of components of $G' - V(P')$ which do not contain b. For any $K \in \mathcal{K}$, let $u_K, u'_K \in V(P')$ such that $N_{G'}(K) \cap V(P') \subseteq V(P'[u_K, u'_K])$ and $P'[u_K, u'_K]$ is minimal with respect to this property. If $|\mathcal{K}| \geq 2$, choose $K \in \mathcal{K}$ such that for any $K' \neq K$, if $E(P[u_K, u'_K]) \cap E(P[u_{K'}, u'_{K'}]) \neq \emptyset$, then $P[u_K, u'_K] \subseteq P[u_{K'}, u'_{K'}]$; such a component must exist because of planarity. If $|\mathcal{K}| = 1$, let $\mathcal{K} = \{K\}$. In either case, $N_G(P'(u_K, u'_K)) \subseteq V(K) \cup \{u_K, u'_K, b'\}$. Thus, $K \cup P'(u_K, u'_K)$ is contained in a component of $G - \{u_K, u'_K, b'\}$ that does not contain any vertex in $\{a, a', b, b'\}$, which contradicts the assumption that G is $(4, \{a, a', b, b'\})$-connected.

So $G' - V(P') = G - (V(P') \cup \{b'\})$ is connected. Hence, $G - V(P')$ has exactly two components K and K' with $b \in V(K)$ and $b' \in V(K')$.

We now show that P' is an induced path in G. Suppose on the contrary that P' is not induced. Let $e = xy \in E(G) - E(P')$ with $x, y \in V(P')$. Then $V(P'(x, y)) \neq \emptyset$. Moreover, by planarity $N_G(P'(x, y)) \subseteq \{x, y, b'\}$. Then $P'(x, y)$ is contained in a component of $G - \{x, y, b'\}$ that does not contain any vertex in $\{a, a', b, b'\}$, which contradicts again the assumption that G is $(4, \{a, a', b, b'\})$-connected.

Thus, P' is a path as required. Moreover, it is easy to see that such a path can be found in $O(|V(G)| + |E(G)|)$ time. □

We conclude this section with another lemma which concerns nonseparating induced paths in planar graphs.

LEMMA 2.7. *Let (G, a, a', b, b') be a planar graph with $|V(G)| \geq 5$ and suppose G is $(4, \{a, a', b, b'\})$-connected and $G \ncong K_{1,4}$. Then there exists a nonseparating induced a-a' path P' in G such that $V(P') \cap \{b, b'\} = \emptyset$. Moreover, such a path can be found in $O(|V(G)| + |E(G)|)$ time.*

Proof. For convenience, let $S := \{a, a', b, b'\}$. Take an embedding of G in a closed disk such that a, a', b, b' occur on the boundary of the disk in the cyclic order listed. By Lemma 2.5, this can be done in $O(|V(G)| + |E(G)|)$ time. Let $G' := G + \{ab', a'b\}$.

We claim that G' is 2-connected. Suppose for a contradiction that G' is not 2-connected. Let x be a cut vertex of G'. Since $|V(G)| \geq 5$ and G (and hence, G') is $(4, S)$-connected, it follows that any component of $G' - x$ either contains vertices only in S or contains at least one vertex in $V(G) - S$ and at least three vertices in S. Since $a'b, ab' \in E(G')$, $G' - x$ cannot have both kinds of components. Therefore, every component of $G' - x$ contains vertices only in S. Moreover, since $|V(G)| \geq 5$, $x \notin S$. But then, it is easy to see that (G, a, a', b, b') must be isomorphic to $K_{1,4}$ with x as the vertex of degree four, which contradicts the hypothesis. Hence, G' is 2-connected.

Thus, we can assume that $ab', a'b$ are in the cycle bounding the infinite face of G'. Let P' be the a-a' subpath of this cycle which avoids b and b'. Note that P' is an a-a' path in G and such a path can be found in $O(|V(G)| + |E(G)|)$ time.

We claim that P' is nonseparating in G. Suppose for a contradiction that $G' - V(P')$ is not connected. Note that b and b' are contained in a component of $G - V(P')$. Let $\mathcal{K}$ be the set of components of $G' - V(P')$ which contain neither b nor b'. For any $K \in \mathcal{K}$, let $u_K, u'_K \in V(P')$ such that $N_{G'}(K) \cap V(P') \subseteq V(P'[u_K, u'_K])$ and $P'[u_K, u'_K]$ is minimal with respect to this property. If $|\mathcal{K}| \geq 2$, choose $K \in \mathcal{K}$ such that for any $K' \neq K$, if $E(P[u_K, u'_K]) \cap E(P[u_{K'}, u'_{K'}]) \neq \emptyset$, then $P[u_K, u'_K] \subseteq P[u_{K'}, u'_{K'}]$; such a component must exist because of planarity. If $|\mathcal{K}| = 1$, let $\mathcal{K} = \{K\}$. In either case, $N_G(P'(u_K, u'_K)) \subseteq V(K) \cup \{u_K, u'_K\}$. Thus, $K \cup P'(u_K, u'_K)$ is contained in a component of $G - \{u_K, u'_K\}$ that does not contain any vertex in S, which contradicts the assumption that G is $(4, S)$-connected. Thus, $G - V(P')$ is connected.

Next we show that P' is an induced path in G. Suppose by contradiction that P' is not induced. Let $e = xy \in E(G) - E(P')$ such that $x, y \in V(P')$. Then $V(P'(x, y)) \neq \emptyset$. Moreover, by planarity $N_G(P'(x, y)) \subseteq \{x, y\}$. Then $P'(x, y)$ is contained in a component of $G - \{x, y\}$ that does not contain any vertex in S, which again contradicts the assumption that G is $(4, S)$-connected.

Thus, P' is a nonseparating induced a-a' path in G such that $V(P') \cap \{b, b'\} = \emptyset$ as required. □

3. Internal chains.

In this section, we prove the following theorem, which will be used to construct internal chains in a nonseparating chain decomposition. See Figure 3 for an illustration of the statement of the result. Recall that, for a graph K and $u, v \in V(K)$, $K - uv$ denotes the graph with vertex set $V(K)$ and edge set $E(K) - \{uv\}$ (note that uv need not be an edge of K).

DEFINITION 3.1. *Let G be a 4-connected graph, let F be a subgraph of G, and let $r \in V(F)$ such that $G_F := G - (V(F) - \{r\})$ is 2-connected. For any distinct $a, a' \in V(F)$, an a-a' path in $G - aa'$ is said to be a* feasible *F-path if the following hold:*

(i) *$V(P) \cap V(F) = \{a, a'\}$ and P is an induced path in $G - aa'$;*

(ii) *$P(a, a')$ is a non-separating path in G_F;*

(iii) *r is contained in a nontrivial block B_P of $G_F - V(P(a, a'))$; and*

(iv) *if $r \in \{a, a'\}$, then r is not a cut vertex of $G_F - V(P(a, a'))$.*

Remark 1. Condition (iv) in Definition 3.1 is necessary for a technical reason, and the reader may want to assume in a first reading that $r \notin \{a, a'\}$ to become familiar with the proof of the next result.

THEOREM 3.2. *Let G be a 4-connected graph, let F be a subgraph of G, and let $r \in V(F)$ such that $G_F := G - (V(F) - \{r\})$ is 2-connected. Suppose that G has a* feasible *a-a'* F-path *P for some $a, a' \in V(F)$. Then there exists a good F-chain*

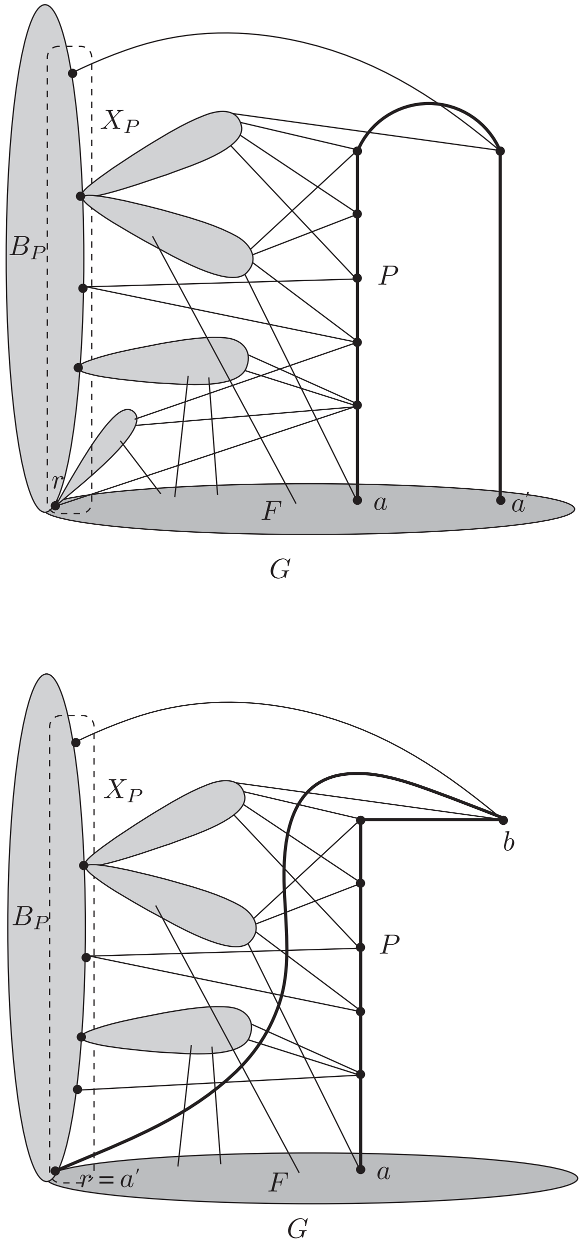


FIG. 3. *Illustration for Theorem* 3.2 *and Notation and Definition* 3.3*: one with* $r \notin \{a, a'\}$ *and the other with* $r \in \{a, a'\}$.

H in G such that $G_F - I(H)$ is 2*-connected, $G[V(F) \cup I(H)]$ is* 2*-connected, and $B_P \subseteq G_F - I(H)$. Moreover, such a chain can be found in $O(|V(G)||E(G)|)$ time.*

Throughout the rest of this section, we fix the following notation.

NOTATION AND DEFINITION 3.3. *Let G be a* 4*-connected graph, let F be a subgraph of G, and let $r \in V(F)$ such that $G_F := G - (V(F) - \{r\})$ is* 2*-connected.*

Suppose G has a feasible a-a' F-path P and r is contained in a nontrivial block B_P of $G_F - V(P(a,a'))$.

*Let $\mathcal{P}_P$ be the set of feasible F-paths P' (with ends, say u, u') in G such that $B_P \subseteq G_F - V(P'(u,u'))$. For each $P' \in \mathcal{P}_P$ with ends, say u, u', let $B_{P'}$ denote the block of $G_F - V(P'(u,u'))$ which contains B_P. We say that $P' \in \mathcal{P}_P$ is a B_P-*augmenting path *if $|V(B_P)| < |V(B_{P'})|$.*

We will describe an algorithm for finding a good F-chain as required in Theorem 3.2. The idea of the algorithm is roughly the following. At the beginning of each iteration we have vertices $a, a' \in V(F)$ and a feasible a-a' F-path P in G. The algorithm iteratively tries to find a B_P-augmenting path P' with ends u, u', and start a new iteration with u, u', P' as a, a', P, respectively. Note that r, u, u', P', F and G (as r, a, a', P, F and G, respectively) satisfy the hypotheses of Theorem 3.2 with B_P enlarged to $B_{P'}$. When the algorithm does not find such a path, it finds a good F-chain as required in Theorem 3.2.

The next lemma says that (assuming G has a feasible a-a' F-path P) one can find in $O(|V(G)| + |E(G)|)$ time a feasible u-u' F-path P' such that $|V(P')| = 3$ or $N_G(P'(u,u')) \cap V(F) \subseteq \{u, u'\} \cup \{r\}$. The latter condition is equivalent to requiring that $N_G(P'(u,u')) \cap V(F) = \{u, u'\}$ when $r \in \{u, u'\}$ or that $N_G(P'(u,u')) \cap V(F) \subseteq \{u, u', r\}$ when $r \notin \{u, u'\}$ (see Figure 3).

LEMMA 3.4. *There exist $u, u' \in V(F)$ and a feasible u-u' F-path P' such that*

(1) $|V(P')| = 3$ *or* $N_G(P(u,u')) \cap V(F) \subseteq \{u, u'\} \cup \{r\}$, *and*

(2) $B_P \subseteq B_{P'}$.

Moreover, such a path can be found in $O(|V(G)| + |E(G)|)$ time.

Proof. If either $|V(P)| = 3$ or $N_G(P(a,a')) \cap V(F) \subseteq \{a, a'\} \cup \{r\}$, then the result follows with $P' := P$.

Thus, assume that $|V(P)| \geq 4$ and $(N_G(P(a,a')) \cap V(F)) - (\{a, a'\} \cup \{r\}) \neq \emptyset$. By symmetry, we may assume that $a \neq r$. Let $v \in V(P(a,a'))$ such that v has a neighbor in $V(F) - (\{a, a'\} \cup \{r\})$, and subject to this, $P[a, v]$ is minimal. If v has two neighbors in $V(F) - \{r, a\}$, say u and u', let $P' := (u, v, u')$. In this case, (1) holds with $|V(P')| = 3$. If v has exactly one neighbor in $V(F) - \{r, a\}$, say u, then let $P' := P[a, v] + \{u, vu\}$ and $u' := a$. Note that in both cases $r \notin \{u, u'\}$. By the choice of v, $N_G(P(u,u')) \cap V(F) \subseteq \{u, u'\} \cup \{r\}$, and hence, (1) holds. Moreover, since $G_F - V(P(a,a')) \subseteq G_F - V(P'(u,u'))$, we have $B_P \subseteq G_F - V(P'(u,u'))$, and hence, (2) holds.

Finally, we show that P' is a feasible u-u' F-path. Since P is induced in $G - aa'$, P' is induced in $G - uu'$. Clearly $V(P') \cap V(F) = \{u, u'\}$, so (i) of Definition 3.1 holds. Since G_F is 2-connected and $P(a,a')$ is an induced path in $G_F - aa'$, if $V(P(v,a')) \neq \emptyset$, then $N_{G_F}(P(v,a')) \cap (V(G_F) - V(P(a,a'))) \neq \emptyset$. Thus, since $G_F - V(P(a,a'))$ is connected, $P'(u,u')$ is nonseparating in G_F, so (ii) of Definition 3.1 holds. Also, r is contained in a nontrivial block of $G_F - V(P'(u,u'))$ because $r \in B_P \subseteq G_F - V(P'(u,u'))$, so (iii) of Definition 3.1 holds. Since $r \notin \{u, u'\}$, we do not need to verify (iv) of Definition 3.1.

Therefore, P' is a feasible F-path as required, and it is not hard to see that such a path P' can be found in $O(|V(G)| + |E(G)|)$ time. □

Assumption 1. Using Lemma 3.4, we can preprocess a feasible F-path at the beginning of each iteration (in $O(|V(G)| + |E(G)|)$ time). Henceforth, we may assume that for the (current) feasible F-path P, $|V(P)| = 3$ or $N_G(P(a,a')) \cap V(F) \subseteq \{a, a'\} \cup \{r\}$. We may also assume that $G_F - V(P(a,a'))$ is not 2-connected; otherwise, $H := P$ gives an F-chain as required in Theorem 3.2: H is an up F-chain (where each of its blocks is trivial), or H is an elementary F-chain. Moreover, $G_F - I(H) = G_F - V(P(a,a'))$ is 2-connected.

NOTATION 3.5. *Let $X_P := N_{G_F}(G_F - V(B_P))$. For each B_P-bridge B of $G_F - V(P(a,a'))$, let r_B denote the unique vertex in $V(B) \cap V(B_P)$. Note that $r_B \in X_P$. Also, if $r \in \{a,a'\}$, then $r \in X_P$.*

Remark 2. Note that since G_F is 2-connected, we have $|X_P| \geq 2$. Moreover, if B is a B_P-bridge of $G_F - V(P(a,a'))$, then $V(B) - \{r_B\}$ has a neighbor in $V(P(a,a'))$.

The next lemma shows that if, for every B_P-bridge B of $G_F - V(P(a,a'))$, $N_G(B - r_B) \subseteq V(P)$, then one can find efficiently a good F-chain (in fact, an up F-chain) H as required in Theorem 3.2 by invoking Theorem 1.7.

LEMMA 3.6. *Suppose that for every B_P-bridge B of $G_F - V(P(a,a'))$, $N_G(B - r_B) \subseteq V(P)$. Then there exists an a-a' up F-chain H in G such that $G_F - I(H)$ is 2-connected, $G[V(F) \cup I(H)]$ is 2-connected, and $B_P \subseteq G_F - I(H)$. Moreover, such a chain can be found in $O(|V(G)||E(G)|)$ time.*

Proof. Suppose first that $r \notin \{a,a'\}$ (see Figure 3). Let G' be the graph obtained from G_F by adding $\{a,a'\}$ and the edges of G from $\{a,a'\}$ to $V(G_F) - \{r\}$. Note that P is a nonseparating induced a-a' path in G'. Note also that B_P is a nontrivial block of $G' - V(P)$. Let $X'_P = N_{G'}(G' - V(B_P))$.

We claim that $G' - (V(B_P) - X'_P)$ is $(4, X'_P \cup \{a,a'\})$-connected. For convenience, let $K := G' - (V(B_P) - X'_P)$. Since, for any B_P-bridge B of $G' - V(P) = G_F - V(P(a,a'))$, $V(B) - \{r_B\}$ has a neighbor in $V(P(a,a'))$, it follows that K is connected and $K - (X'_P \cup \{a,a'\})$ is a component of $G - (X'_P \cup \{a,a'\})$. Hence, because G is 4-connected, K is $(4, X'_P \cup \{a,a'\})$-connected.

Thus, the hypotheses of Theorem 1.7 are satisfied with G', a, a', P, B_P, X'_P as G, a, b, P, B_P, X_P, respectively. Hence, there exists a planar a-a' chain H in G' such that $G' - V(H) = G_F - I(H)$ is 2-connected and $B_P \subseteq G' - V(H) = G_F - I(H)$. Moreover, such a chain can be found in $O(|V(G')||E(G')|)$ time (and hence, in $O(|V(G)||E(G)|)$ time). Note also that H is an up F-chain in G. Hence, $G[V(F) \cup I(H)]$ is 2-connected, so the result follows.

Now suppose that $r \in \{a,a'\}$, and without loss of generality, let $r = a'$ (see Figure 3). Let b be the neighbor of r in P. Let G' be the graph obtained from G_F by adding a and the edges of G from a to $V(G_F) - \{r\}$. Note that $b \in V(G')$ and $P[a,b]$ is a nonseparating induced path in G'. Note also that B_P is a nontrivial block of $G' - V(P[a,b]) = G_F - V(P(a,r))$. Let $X'_P = N_{G'}(G' - V(B_P))$. Since P is a feasible a-r F-path, r is not a cut vertex of $G' - V(P[a,b]) = G_F - V(P(a,r)$ (in particular, there is no B_P-bridge in $G' - V(P[a,b])$ containing r).

We claim that $G' - (V(B_P) - X'_P)$ is $(4, X'_P \cup \{a,b\})$-connected. For convenience, let $K := G' - (V(B_P) - X'_P)$. Since, for any B_P-bridge B of $G_F - V(P(a,r))$, $V(B) - \{r_B\}$ has at least two neighbors in $V(P(a,r))$ (because G is 4-connected), it follows that $V(B) - \{r_B\}$ has at least one neighbor in $V(P(a,b))$. Hence, K is connected and $K - (X'_P \cup \{a,b\})$ is a component of $G - (X'_P \cup \{a,b\})$. Since G is 4-connected, K is $(4, X'_P \cup \{a,b\})$-connected.

Thus, the hypotheses of Theorem 1.7 are satisfied with $G', a, b, P[a,b], B_P, X'_P$ as G, a, b, P, B_P, X_P, respectively. Hence, there exists a planar a-b chain H' in G' such that $G' - V(H')$ is 2-connected and $B_P \subseteq G' - V(H')$. Moreover, such a chain can be found in $O(|V(G')||E(G')|)$ time (and hence, $O(|V(G)||E(G)|)$ time). Since b is the only neighbor of r in $V(P) - \{a,r\}$ and no B_P-bridge in $G' - V(P[a,b])$ contains r, $r \notin N_G(V(H') - \{a,b\})$. Thus, $H := H' + rb$ is an up a-r F-chain in G (recall

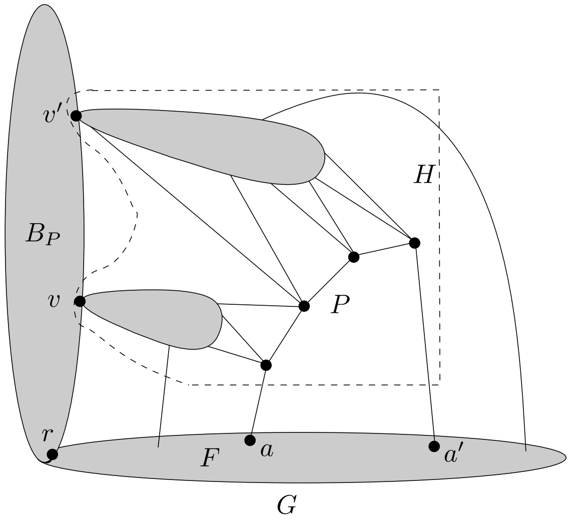


FIG. 4. *Graph H in the proof of Lemma* 3.7.

$a' = r$), so $G[V(F) \cup I(H)]$ is 2-connected. Note also that $G_F - I(H) = G' - V(H')$ is 2-connected, and hence, the result follows. □

Next, we show that if $|X_P| = 2$, then one can find efficiently either a B_P-augmenting path or a good F-chain as required in Theorem 3.2.

LEMMA 3.7. *Suppose that $|X_P| = 2$, and let v, v' be the vertices in X_P. Then exactly one of the following holds:*

(1) *there exists a B_P-augmenting path; or*

(2) *$H := (G_F - (V(B_P) - X_P)) - vv'$ is a down v-v' F-chain in G such that $G_F - I(H)$ is 2-connected and $G[V(F) \cup I(H)]$ is 2-connected.*

Moreover, one can in $O(|V(G)| + |E(G)|)$ time either find a path as in (1) *or certify that* (2) *holds.*

Proof. Let $H := (G_F - (V(B_P) - X_P)) - vv'$. Since G_F is 2-connected and $X_P = \{v, v'\}$, H is a v-v' chain in G and $N_G(H - \{v, v'\}) \subseteq V(F - r) \cup \{v, v'\}$. See Figure 4 for an example. Let $H := v_0B_1v_1 \ldots v_{k-1}B_kv_k$, where $v_0 = v$ and $v_k = v'$. This decomposition of H into blocks can be computed in $O(|V(G)| + |E(G)|)$ time. If every block of H is trivial, then H is a down F-chain, $G_F - I(H) = B_P$ is 2-connected, and $G[V(F) \cup I(H)]$ is 2-connected, so (2) holds.

Thus, we may assume that H contains a nontrivial block. For each nontrivial block B_i, let $S_i := V(F - r) \cap N_G(B_i - \{v_{i-1}, v_i\})$, and let G_i be the graph obtained from B_i by adding S_i and the edges of G from S_i to $V(B_i) - \{v_{i-1}, v_i\}$. Note that $G_i - S_i = B_i$ is 2-connected and $B_i - \{v_{i-1}, v_i\}$ is a union of components of $G - (S_i \cup \{v_{i-1}, v_i\})$. Because G is 4-connected, G_i is $(4, S_i \cup \{v_{i-1}, v_i\})$-connected, and every component of $B_i - \{v_{i-1}, v_i\}$ has at least two neighbors in S_i. Thus, the hypotheses of Lemma 2.3 are satisfied with G_i, S_i, v_{i-1}, v_i as G, S, b, b', respectively.

Hence, either (a) there exist $u_i, u'_i \in S_i$ and an induced u_i-u'_i path P'_i in G_i such that $V(P_i) \cap \{v_{i-1}, v_i\} = \emptyset$, $V(P_i) \cap S_i = \{u_i, u'_i\}$, and $G_i - (V(P_i) \cup S_i)$ is connected, or (b) $|S_i| = 2$ and the elements of S_i can be labeled as u_i, u'_i such that $(G_i, v_{i-1}, u_i, v_i, u'_i)$ is planar. Moreover, one can in $O(|V(G_i)| + |E(G_i)|)$ time find a path as in (a) or certify that (b) holds. If (a) holds for some nontrivial block B_i, then P'_i is a B_P-augmenting path for the following reasons: (i)–(iii) of Definition 3.1 hold,

$r \notin \{u, u'\}$ (so (iv) of Definition 3.1 holds), and there exists a v-v' path contained in $H - V(P_i'(u_i, u_i'))$ (so B_P is properly contained in $B_{P_i'}$). If (b) holds for every nontrivial block B_i, then H is clearly a down F-chain, $G[V(F) \cup I(H)]$ is 2-connected (because G is 4-connected, and so $G_i - \{v_{i-1}, v_i\}$ is a u_i-u_i' chain), and $G_F - I(H)$ is 2-connected.

One can verify that either (1) or (2) holds in $O(|V(G)| + |E(G)|)$ time because if (b) holds for a nontrivial block B_i, then $|V(G_i)| + |E(G_i)| = O(|V(B_i)| + |E(B_i)|)$, and if (a) holds for some G_i, then $|V(G_i)| + |E(G_i)| = O(|V(G)| + |E(G)|)$. In the latter case, we find a B_P-augmenting path and we stop. Thus, this verification can be carried out in $O(|V(G)| + |E(G)|)$ time. □

The following lemma shows that if $|X_P| \geq 3$ and $|V(P)| = 3$, then one can find efficiently a B_P-augmenting path.

Lemma 3.8. *Suppose that $|X_P| \geq 3$ and $|V(P)| = 3$. Then exactly one of the following holds:*

(1) *there exists a B_P-augmenting path; or*

(2) *P is an elementary F-chain in G such that $G_F - I(P)$ is 2-connected and $G[V(F) \cup I(P)]$ is 2-connected.*

Moreover, one can in $O(|V(G)| + |E(G)|)$ time either find a path as in (1) *or certify that* (2) *holds.*

Proof. If $G_F - V(P(a, a'))$ is 2-connected, then P is an elementary F-chain in G, $G_F - I(P)$ is 2-connected, and $G[V(F) \cup I(P)]$ is 2-connected, so (2) holds. Note, this can be checked in $O(|V(G)| + |E(G)|)$ time.

So we may assume that $G_F - V(P(a, a'))$ is not 2-connected. Let K be a B_P-bridge of $G_F - V(P(a, a'))$, and let v denote the unique vertex in $V(P(a, a'))$. If K is 2-connected, then let $B := K$ and $b := r_K$. Otherwise let B be an endblock of K not containing r_K, and let b denote the cut vertex of K contained in $V(B)$. Since G_F is 2-connected, $v \in N_G(B - b)$. Note that B can be computed in $O(|V(G)| + |E(G)|)$ time.

First, suppose that B is trivial, and let w be the unique vertex in $V(B - b)$. Since G is 4-connected, w has at least three neighbors in $V(F - r) \cup \{v\}$, and hence, it has two neighbors u, u' in $V(F - r)$. Let $P' := (u, w, u')$. We claim that P' is a feasible F-path. Clearly, P' is an induced path in $G - uu'$ and $V(P') \cap V(F) = \{u, u'\}$. Since G_F is 2-connected, $G_F - V(P'(u, u')) = G_F - w$ is connected. Thus, $P'(u, u')$ is nonseparating in G_F. Also $r \in V(B_P)$ and $B_P \subseteq G_F - V(P'(u, u'))$. Therefore, since $r \notin \{u, u'\}$, P' is a feasible F-path. Since $|X_P| \geq 3$, there exists a path (containing v) with ends in $X_P - \{r_B\}$ which is internally disjoint from $V(B_P) \cup V(B)$. Therefore, B_P is properly contained in $B_{P'}$, and hence, P' is a B_P-augmenting path.

Thus, we may assume that B is nontrivial, so B is 2-connected. Let $S := N_G(B - b) - \{b, v\}$, and let G' be obtained from B by adding S and the edges of G from S to $V(B) - \{b\}$. Note that $S \subseteq V(F - r)$ and $G' - S = B$ is 2-connected. Since G is 4-connected, $G[V(G') \cup \{v\}]$ is $(4, S \cup \{b, v\})$-connected, and hence, G' is $(3, S \cup \{b\})$-connected. By Lemma 2.4 (with G', b, S as G, b, S, respectively) there exist $u, u' \in S$ and an induced u-u' path P' in G' such that $V(P') \cap \{b\} = \emptyset$, $V(P') \cap S = \{u, u'\}$, and $G' - (V(P') \cup S)$ is connected. Moreover, such a path can be found in $O(|V(G')| + |E(G')|)$ time (and hence, in $O(|V(G)| + |E(G)|)$ time).

We claim that P' is a feasible F-path. Clearly, P' is an induced path in $G - uu'$ and $V(P') \cap V(F) = \{u, u'\}$. Since $G' - (V(P') \cup S) = B - V(P'(u, u'))$ is connected and $b \notin V(P')$, we have that $G_F - V(P'(u, u'))$ is connected. Thus, $P'(u, u')$ is nonseparating in G_F. Also $r \in V(B_P)$, and $B_P \subseteq G_F - V(P'(u, u'))$. Since $r \notin S$,

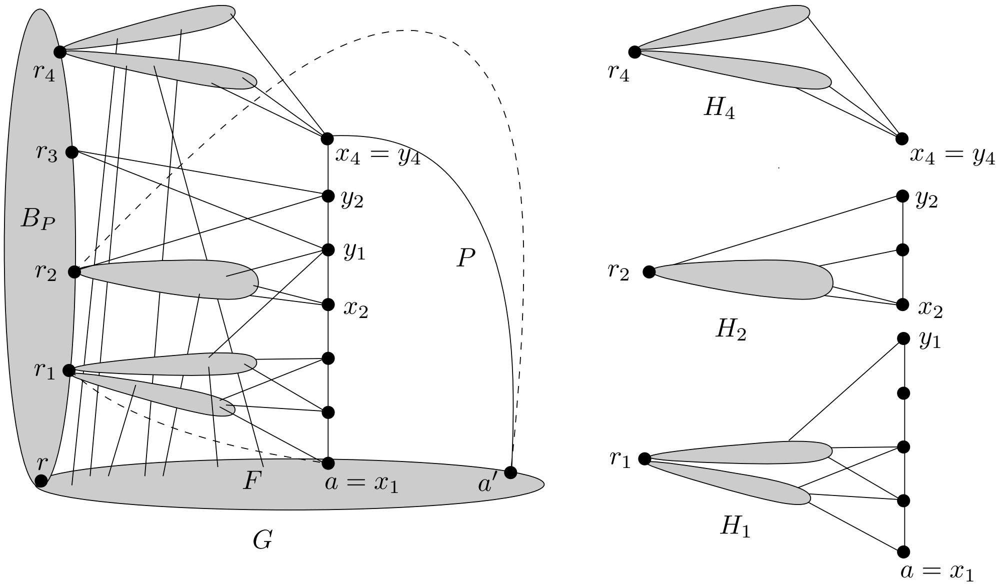


FIG. 5. *Example for Notation* 3.9 *with $X_P = \{r_1, r_2, r_3, r_4\}$. Note that the edges r_1a, r_2a' are not contained in any H_i.*

$r \notin \{u, u'\}$, so P' is a feasible F-path. Furthermore, since $|X_P| \geq 3$, there exists a path (containing v) with ends in $X_P - \{r_B\}$ which is internally disjoint from $V(B_P) \cup V(B)$. Therefore, B_P is properly contained in $B_{P'}$, and hence, P' is a B_P-augmenting path. □

By Lemmas 3.6, 3.7, and 3.8, we need to deal only with the case where $|X_P| \geq 3$, $|V(P)| \geq 4$, and for some B_P-bridge B of $G_F - V(P(a, a'))$, $B - r_B$ has a neighbor in $V(F - r) - \{a, a'\}$. Our aim is to prove that we can find either a B_P-augmenting path or a triangle F-chain H such that $G_F - I(H)$ is 2-connected. In order to do this, we need to introduce some notation and prove auxiliary results.

NOTATION 3.9. *For any $x, y \in V(P)$, we denote $x \leq y$ if $x \in V(P[a, y])$. If $x \leq y$ and $x \neq y$, then we write $x < y$. In this case, we say that x is* lower *than y, or y is* higher *than x.*

Let $X_P := \{r_1, \ldots, r_p\}$. For each i, $1 \leq i \leq p$, if r_i is a cut vertex of $G_F - V(P(a, a'))$, then let $V_i := \bigcup V(B)$, where the union is taken over all the B_P-bridges B of $G_F - V(P(a, a'))$ with $r_B = r_i$; if r_i is not a cut vertex of $G_F - V(P(a, a'))$, then let $V_i := \{r_i\}$.

For each i such that $V_i \neq \{r_i\}$, let $x_i, y_i \in V(P)$ with $x_i \leq y_i$ such that G has an edge from x_i (y_i, respectively) to V_i which is not an edge from $\{a, a'\}$ to r_i, and subject to this, $P[x_i, y_i]$ is maximal. Note that we may have $x_i = a$ or $y_i = a'$, but $r \notin \{x_i, y_i\}$ because B_P is a block of $G_F - V(P(a, a'))$.

Let $P_i := P[x_i, y_i]$, and let H_i be the graph obtained from $G[V_i \cup V(P_i)]$ by removing all edges from $\{a, a'\}$ to r_i. Let $\mathcal{H} := \{H_i : 1 \leq i \leq p, V_i \neq \{r_i\}\}$. We say that $H_i \in \mathcal{H}$ is adjacent *to F if $N_G(V_i - \{r_i\}) \cap (V(F - r) - \{a, a'\}) \neq \emptyset$. See Figure 5 for an example.*

LEMMA 3.10. *Every $H_i \in \mathcal{H}$ is an r_i-x_i (and also an r_i-y_i) chain. Moreover, no vertex of P_i is a cut vertex of H_i, and P_i is contained in an endblock of H_i.*

Proof. Since $G[V_i] = H_i - V(P_i)$ is connected and because H_i has edges from both x_i and y_i to V_i, no vertex of P_i is a cut vertex of H_i, and hence, P_i is contained in a block of H_i. We claim that if B is an endblock of H_i, then $r_i \in V(B)$ or $V(P_i) \subseteq V(B)$ (and hence, we have Lemma 3.10). Suppose for a contradiction that B is an endblock of H_i and B contains neither r_i nor any vertex in $V(P_i)$. Let v be the cut vertex of H_i contained in $V(B)$. Then $B - v$ is a component of $G_F - v$, which is a contradiction, since G_F is 2-connected. Similarly, we can show that H_i is an r_i-y_i chain. □

NOTATION 3.11. *For each $H_i \in \mathcal{H}$ with $x_i \neq y_i$, let A_i denote the block of H_i containing P_i. If $A_i \neq H_i$, then let b_i denote the cut vertex of H_i contained in A_i. If $A_i = H_i$, then let $b_i := r_i$.*

The next lemma illustrates two situations when we can find a B_P-augmenting path.

LEMMA 3.12. *Assume that $|X_P| \geq 3$, and let $H_i \in \mathcal{H}$. Suppose that one of the following holds:*

(i) *$x_i = y_i$; or*

(ii) *$x_i \neq y_i$, and H_i contains at least three blocks or H_i contains a nontrivial block other than A_i.*

Then one can find a B_P-augmenting path in $O(|V(G)| + |E(G)|)$ time.

Proof. If $x_i = y_i$, then let $H := H_i$. If $x_i \neq y_i$, then let $H := H_i - (V(A_i) - \{b_i\})$. Note that H is an r_i-x_i chain if $x_i = y_i$, and H is an r_i-b_i chain if $x_i \neq y_i$. Moreover, since (i) or (ii) holds, H is not induced by an edge.

Let $H := v_0 B_1 v_1 \dots v_{k-1} B_k v_k$ with $v_0 = r_i$, $v_k = x_i$ if $x_i = y_i$, and $v_k = b_i$ if $x_i \neq y_i$. This decomposition of H_i into blocks can be computed in $O(|V(G)|+|E(G)|)$ time.

Case 1. There exists $j \in \{1, \dots, k\}$ such that B_j is nontrivial.

Let $S := N_G(B_j - \{v_{j-1}, v_j\}) - \{v_{j-1}, v_j\}$. Note that $S \subseteq V(F - r) - \{a, a'\}$ because B_P is a block of $G_F - V(P(a, a'))$. Let G' be the graph obtained from B_j by adding S and the edges of G from S to $V(B_j) - \{v_{j-1}, v_j\}$. Note that $G' - S = B_j$ is 2-connected and G' is $(4, S \cup \{v_{j-1}, v_j\})$-connected (because G is 4-connected). Therefore, the hypotheses of Lemma 2.3 are satisfied with G', S, v_{j-1}, v_j as G, S, b, b', respectively. Then by Lemma 2.3 exactly one of the following occurs:

(1) there exist $u, u' \in S$ and an induced u-u' path P' in G' such that $V(P') \cap \{v_{j-1}, v_j\} = \emptyset$, $V(P') \cap S = \{u, u'\}$, and $G' - (V(P') \cup S)$ is connected; or

(2) $|S| = 2$, and the elements of S can be labeled as u, u' such that $(G', v_{j-1}, u, v_j, u')$ is planar.

Moreover, one can in $O(V(G')| + |E(G')|)$ time (and hence, in $O(|V(G)| + |E(G)|)$ time) find a path as in (1) or certify that (2) holds.

Note that since $|X_P| \geq 3$, there exists a path W with ends in $X_P - \{r_i\}$ which is internally disjoint from $V(B_P) \cup V_i$.

Suppose (1) holds. We claim that P' is a feasible F-path. Clearly, $V(P') \cap V(F) = \{u, u'\}$, and P' is an induced path in $G - uu'$. Since $B_j - V(P'(u, u')) = G' - (V(P') \cup S)$ is connected and $v_{j-1}, v_j \notin V(P')$, we have that $G_F - V(P'(u, u'))$ is connected. Thus, $P'(u, u')$ is nonseparating in G_F. Also $r \in V(B_P)$, and $B_P \subseteq G_F - V(P'(u, u'))$. Therefore, since $r \notin \{u, u'\}$, P' is a feasible F-path. Moreover, since W is also a path in $G_F - V(P'(u, u'))$, $B_P \cup W \subseteq B_{P'}$. Therefore, P' is a B_P-augmenting path.

Now assume (2) holds. By Lemma 2.6 one can find in $O(|V(G')| + |E(G')|)$ time (and hence, in $O(|V(G)| + |E(G)|)$ time) an induced u-u' path Q in G' such that $G' - V(Q)$ has exactly two components K, K' with $v_{j-1} \in V(K)$ and $v_j \in V(K')$. We claim that Q is a feasible F-path. Clearly, $V(Q) \cap V(F) = \{u, u'\}$, and Q is an induced path in $G - uu'$. Note that $B - Q(u, u') = G' - V(Q)$ has exactly two components (namely K and K'), there exists a path in H_i from $v_{j-1} \in V(K)$ to $r_i \in X_P$ disjoint from Q, and there exists a path from $v_j \in V(K')$ to X_P in $G_F - V(Q(u,u'))$ (because $|X_P| \geq 2$). It follows that $G_F - V(Q(u,u'))$ is connected. Also $r \in V(B_P)$, and $B_P \subseteq G_F - V(Q(u,u'))$. Since $r \notin \{u,u'\}$, Q is a feasible F-path. Moreover, W is a path in $G_F - V(Q(u,u'))$, and hence, $B_P \cup W \subseteq B_Q$. Therefore, Q is a B_P-augmenting path.

Case 2. All blocks of H are trivial.

By (ii), H_i contains at least two blocks other than A_i, and hence, $k \geq 3$. So B_1 and B_2 are trivial blocks of H. Since G is 4-connected, v_1 has at least two neighbors in $V(F - r)$, say u, u'. Let $P' := (u, v_1, u')$. We claim that P' is a feasible F-path. Clearly, $V(P') \cap V(F) = \{u,u'\}$, and P' is an induced path in $G - uu'$. Since G_F is 2-connected, $G_F - V(P'(u,u')) = G_F - v_1$ is connected. Also since $B_P \subseteq G_F - V(P'(u,u'))$ and $r \notin \{u,u'\}$, it follows that P' is a feasible F-path. Moreover, one can see that $B_P \cup W \subseteq B_{P'}$. Therefore, P' is a B_P-augmenting path. □

Now we study the case where, for every $H_i \in \mathcal{H}$, $x_i \neq y_i$, H_i has at most two blocks, and if H_i has exactly two blocks, then A_i is the only nontrivial block of H_i. We give three lemmas which deal with this case. The arguments used for many cases in the proofs are similar, but unfortunately it seems necessary to cover all of those cases. We frequently produce a B_P-augmenting path P' in the following way. We first exhibit a nontrivial path W in G_F with ends in B_P such that W is internally disjoint from B_P. We then produce a feasible F-path P' disjoint from W such that $V(B_P) \cup V(W) \subseteq V(B_{P'})$, so P' is B_P-augmenting. For the sake of brevity, when we state a result occurs "because of the path W," we are implicitly using this technique.

Recall that by Assumption 1 we may assume that if $|V(P)| \geq 4$, then $N_G(P(a,a')) \cap V(F) \subseteq \{a,a'\} \cup \{r\}$.

LEMMA 3.13. *Assume that $|X_P| \geq 3$, $|V(P)| \geq 4$, and, for every $H_j \in \mathcal{H}$, $x_j \neq y_j$. Suppose that, for every $H_j \in \mathcal{H}$, $V(A_j) - \{b_j, x_j, y_j\}$ has no neighbor in $V(F - r) - \{a, a'\}$. Assume that for some $H_i \in \mathcal{H}$, H_i is adjacent to F. Then exactly one of the following holds:*

(1) *there exists a B_P-augmenting path; or*

(2) *there exists a triangle F-chain H in G such that $I(H) = V(G_F) - V(B_P)$, $G_F - I(H)$ is 2-connected, and $G[V(F) \cup I(H)]$ is 2-connected.*

Moreover, one can in $O(|V(G)|+|E(G)|)$ time find either a path as in (1) *or a triangle F-chain as in* (2).

Proof. Let us first show that (1) and (2) are mutually exclusive. Suppose that (2) holds. It is not hard to see that there exists no B_P-augmenting path because every feasible F-path must use exactly two vertices of $V(G_F) - V(B_P)$. Thus, it remains to show that either (1) or (2) holds and that one can determine in $O(|V(G)| + |E(G)|)$ time which of them occurs.

We consider two cases.

Case 1. There exist distinct $m, n \in \{1, \ldots, p\} - \{i\}$ such that both V_m and V_n have a neighbor in $V(P(x_i, a'))$ or both V_m and V_n have a neighbor in $V(P(a, y_i))$.

Without loss of generality, assume that both V_m and V_n have a neighbor in $V(P(x_i, a'))$.

We claim that A_i contains a nonseparating induced b_i-x_i path Q such that $V(Q) \cap (V(P_i) - \{x_i\}) = \emptyset$. This is obvious if $V(A_i) - V(P_i) = \{b_i\}$ because then b_i must be adjacent to x_i, and the result follows by taking Q as the path induced by the edge $b_i x_i$. Thus, we may assume that $V(A_i) - V(P_i) \neq \{b_i\}$. Let S_i denote the set of vertices in $V(P(x_i, y_i))$ which have a neighbor in $(\bigcup_{j=1}^{p} V_j) - V_i$. Since G is 4-connected, A_i

is $(4, S_i \cup \{b_i, x_i, y_i\})$-connected. Moreover, $A_i - (V(P_i) - \{x_i\})$ is connected and $S_i \cup \{y_i\} \subseteq V(P_i) - \{x_i\}$, so there exists a b_i-x_i path Q' in A_i such that $V(P_i) - \{x_i\}$ (and hence, $S_i \cup \{y_i\}$) is contained in a component U of $A_i - V(Q')$. Therefore, the hypotheses of Lemma 2.1 are satisfied with $A_i, S_i \cup \{b_i, x_i, y_i\}, b_i, x_i, Q', U$ as G, S, a, a', P, U, respectively. By Lemma 2.1 one can find in $O(|V(G)| + |E(G)|)$ time a nonseparating induced b_i-x_i path Q in A_i such that $V(Q) \cap V(U) = \emptyset$. Since $V(P_i) - \{x_i\} \subseteq V(U)$, we have $V(Q) \cap (V(P_i) - \{x_i\}) = \emptyset$, and thus, Q is a path as required.

By hypothesis, $x_i \neq y_i$, so by Lemma 3.12, we can in $O(|V(G)| + |E(G)|)$ time either find a B_P-augmenting path or certify that H_i has at most two blocks. Hence, we may assume that H_i has at most two blocks. Since H_i is adjacent to F and $V(A_i) - \{b_i, x_i, y_i\}$ has no neighbors in $V(F - r) - \{a, a'\}$, it follows that H_i has exactly two blocks, and b_i is adjacent to some vertex $u \in V(F - r) - \{a, a'\}$. Let $P' := (Q \cup P[a, x_i]) + \{u, b_i u\}$. By assumption, both V_m and V_n have a neighbor on $P(x_i, a')$. Since P' is disjoint from $V(P_i) - \{x_i\}$, there exists an r_m-r_n path W in $G_F - V(P'(a, u))$ which is internally disjoint from $V(B_P) \cup V_i \cup \{x_i\}$.

Next we show that P' is a B_P-augmenting path. Since $N_G(P(a, a')) \cap V(F) \subseteq \{a, a'\} \cup \{r\}$ (by Assumption 1) and P is induced in $G - aa'$, we have that P' is an induced u-a path in $G - au$. Also, since $A_i - V(Q)$ is connected, $P'(a, u)$ is nonseparating in G_F. Note also that if r is an end of P', then $a = r$, and r is not a cut vertex of $G_F - V(P(a, a'))$. Then, because of the path W, r is not a cut vertex of $G_F - V(P'(a, u))$. Thus, P' is a feasible F-path. Since $B_P \cup W \subseteq B_{P'}$, P' is a B_P-augmenting path and (1) holds.

Case 2. For any distinct $m, n \in \{1, \ldots, p\} - \{i\}$, V_m and V_n do not both have a neighbor in $V(P(x_i, a'))$, nor do both V_m and V_n have a neighbor in $V(P(a, y_i))$.

By hypothesis, $x_i \neq y_i$, so by Lemma 3.12, we can in $O(|V(G)| + |E(G)|)$ time either find a B_P-augmenting path or certify that H_i has at most two blocks. Hence, we may assume that H_i has at most two blocks. Since H_i is adjacent to F and $A_i - \{b_i, x_i, y_i\}$ has no neighbor in $V(F - r) - \{a, a'\}$, it follows that H_i has exactly two blocks, and b_i has at least one neighbor in $V(F - r) - \{a, a'\}$. Moreover, since we are in Case 2, we must have $|X_P| = 3$. Without loss of generality, we may assume that $i = 3$, V_1 has a neighbor in $V(P(a, x_3])$, and V_2 has a neighbor in $V(P[y_3, a'))$. Moreover, V_1 has no neighbor in $V(P(x_3, a'))$, and V_2 has no neighbor in $V(P(a, y_3))$.

Suppose b_3 has two neighbors in $V(F - r) - \{a, a'\}$, say u, u'. Let $P' := (u, b_3, u')$. Clearly, $G_F - V(P'(u, u')) = G_F - b_3$ is connected. Since $r \notin \{u, u'\}$, it is not hard to see that P' is a feasible F-path. Moreover, there exists an r_1-r_2 path which is internally disjoint from $V(B_P) \cup V_i$. Hence, P' is a B_P-augmenting path, and (1) holds. Clearly, P' can be found in $O(|V(G)| + |E(G)|)$ time.

Thus, we may assume that b_3 has exactly one neighbor in $V(F - r) - \{a, a'\}$. We consider two subcases.

Subcase 2.1. For some $j \in \{1, 2\}$, say $j = 1$, $V_1 \neq \{r_1\}$.

Let $H_1 := w_0 B'_1 w_1 \ldots w_{s-1} B'_s w_s$ where $w_0 = r_1$, and $B'_s = A_1$. Since $x_1 \neq y_1$ (by assumption), then from Lemma 3.12 either $s = 1$ or $s = 2$ and B'_1 is trivial.

We claim that $V(A_1) = \{b_1, x_1, y_1\}$. Suppose for a contradiction that $V(A_1) - \{b_1, x_1, y_1\} \neq \emptyset$. Then $A_1 - \{b_1, x_1, y_1\}$ is a component of $G - \{b_1, x_1, y_1\}$ for the following reasons: $V(A_1) - \{b_1, x_1, y_1\}$ has no neighbor in $V(F - r) - \{a, a'\}$ (by hypothesis), $V(P(x_1, y_1))$ has no neighbor in $V_3 \cup V_2$ (by assumption in Case 2), and

P is an induced path in $G-aa'$. But then $\{b_1, x_1, y_1\}$ is a 3-cut in G which contradicts the assumption that G is 4-connected. Thus, $V(A_1) = \{b_1, x_1, y_1\}$.

Therefore, $\{b_1, x_1, y_1\}$ induces a triangle in G. Since $H_1 \in \mathcal{H}$, $V_1 \neq \{r_1\}$. This implies that $s = 2$ and B_1' is a trivial block of H_1 (and hence, r_1 is adjacent to b_1). Since b_1 has degree at least four in G, b_1 must have some neighbor in $V(F-r)$. Hence, H_1 is adjacent to F, and V_2 and V_3 have neighbors in $V(P(x_1, a'))$, so we can proceed as in Case 1 and find a B_P-augmenting path in $O(|V(G)| + |E(G)|)$ time.

Subcase 2.2. For every $j \in \{1, 2\}$, $V_j = \{r_j\}$.

Thus, r_1 has a neighbor in $V(P(a, x_3])$, and hence, $x_3 \neq a$. Similarly, $y_3 \neq a'$.

We claim that $V(A_3) = \{b_3, x_3, y_3\}$. Suppose for a contradiction that $V(A_3) - \{b_3, x_3, y_3\} \neq \emptyset$. Then $A_3 - \{b_3, x_3, y_3\}$ is a component of $G - \{b_3, x_3, y_3\}$ for the following reasons: $V(A_3) - \{b_3, x_3, y_3\}$ has no neighbor in $V(F - r) - \{a, a'\}$ (by hypothesis), $V(P(x_3, y_3))$ has no neighbor in $V_1 \cup V_2$ (by assumption in Case 2), and P is an induced path in $G - aa'$. But then $\{b_3, x_3, y_3\}$ is a 3-cut in G, which contradicts the assumption that G is 4-connected. Thus, $V(A_3) = \{b_3, x_3, y_3\}$, and A_3 is a triangle.

Since G_F is 2-connected and P is an induced path in $G - aa'$, and because $N_G(P(a, a')) \cap V(F) \subseteq \{a, a'\} \cup \{r\}$, it follows that $V(P) = V(P_3) \cup \{a, a'\}$, r_1 is adjacent to x_3, and r_2 is adjacent to y_3. Let u denote the only neighbor of b_3 in $V(F-r)-\{a, a'\}$. Note that $a \neq r$; otherwise $r_1 = r$ (because $|X_P| = 3$), and x_3 would have degree three in G which is a contradiction because G is 4-connected. Similarly, $a' \neq r$. If $r = r_1$, then (r_1, x_3, a) is a B_P-augmenting path. If $r = r_2$, then (r_2, y_3, a') is a B_P-augmenting path. If $r = r_3$, then (r_3, b_3, u) is a B_P-augmenting path. Thus, we may assume that $r \notin \{r_1, r_2, r_3\}$. Therefore, $H := A_i + \{u, a, a', b_iu, x_ia, y_ia'\}$ is a triangle F-chain in G with $b_3, x_3, y_3, u, a, a', r_3, r_1, r_2$ as $v_1, v_2, v_3, u_1, u_2, u_3, w_1, w_2, w_3$, respectively, in Definition 1.2. It is easy to see that $G_F - I(H) = B_P$ is 2-connected and $G[V(F) \cup I(H)]$ is 2-connected. So (2) holds. □

LEMMA 3.14. *Assume that* $|X_P| \geq 3$, $|V(P)| \geq 4$, *and for every* $H_j \in \mathcal{H}$, $x_j \neq y_j$. *Suppose that* $H_i \in \mathcal{H}$ *and* $V(A_i) - \{b_i, x_i, y_i\}$ *has a neighbor in* $V(F - r) - \{a, a'\}$. *Assume that* $V(P(x_i, y_i))$ *has no neighbor in* $(\bigcup_{j=1}^{p} V_j) - V_i$. *Then a* B_P*-augmenting path can be found in* $O(|V(G)| + |E(G)|)$ *time.*

Proof. Since G_F is 2-connected and $V(P(x_i, y_i))$ has no neighbor in $(\bigcup_{j=1}^{p} V_j) - V_i$, there exists $m \in \{1, \ldots, p\} - \{i\}$ such that V_m has a neighbor in $V(P(a, x_i])$ or in $V(P[y_i, a'))$.

By symmetry we may assume that V_m has a neighbor in $V(P[y_i, a'))$. Then $y_i \neq a'$.

First, we find an endblock of $A_i - \{x_i, y_i\}$ in $O(|V(G)| + |E(G)|)$ time as follows. If $A_i - \{x_i, y_i\}$ is 2-connected, then let $B := A_i - \{x_i, y_i\}$, and let $b := b_i$. Otherwise, let B be an endblock of $A_i - \{x_i, y_i\}$, and let b denote the cut vertex of $A_i - \{x_i, y_i\}$ contained in B so that $b_i \notin V(B - b)$. Note that $V(P(x_i, y_i))$ has no neighbors in $(\bigcup_{j=1}^{p} V_j) - V_i$, and $N_G(B - b) \subseteq V(F - r) \cup \{x_i, y_i, b\}$. Since $r \notin \{x_i, y_i\}$ (by the definition of x_i, y_i in Notation 3.9), $r \notin N_G(B - b) - \{b\}$. Moreover, since G is 4-connected, $|N_G(B - b)| \geq 4$. Note that such an endblock B can be found in $O(|V(G)| + |E(G)|)$ time.

Next, we consider two cases.

Case 1. y_i has a neighbor in $V(A_i) - (\{x_i, y_i\} \cup V(B - b))$.

Then, since V_m has a neighbor in $V(P[y_i, a'))$, there exists an r_i-r_m path W in $G_F - V(P(a, x_i])$ which is internally disjoint from $V(B_P)$ and intersects $P[y_i, a')$.

Subcase 1.1. B is trivial.

Let v denote the unique vertex in $V(B)-\{b\}$. Then $N_G(v) \subseteq V(F-r)\cup\{x_i, y_i, b\}$. Since G is 4-connected, v has at least three neighbors in $V(F-r)\cup\{x_i, y_i\}$, and hence, it has two neighbors in $V(F-r)\cup\{x_i\}$. Let u, u' be distinct neighbors of v in $V(F-r)\cup\{x_i\}$, and assume that $u \neq x_i$. By the definition of x_i, y_i in Notation 3.9, one can see that $\{u, u'\}\cap\{a'\} = \emptyset$ and $u \neq a$ (because $y_i \neq a'$ and $x_i \neq u$). If $u' \neq x_i$, then let $P' := (u, v, u')$; otherwise, let $P' := P[a, x_i] + \{u, v, uv, vx_i\}$. Clearly, P' is a path with ends in $V(F)$ which is internally disjoint from $V(B_P)\cup V(F)$.

Next we show that P' is a B_P-augmenting path. Let u, u'' denote the ends of P'. By assumption, $N_G(P(a, a'))\cap V(F) \subseteq \{a, a'\}\cup\{r\}$ (Assumption 1), and P is induced in $G - aa'$. Then since $N_G(v) \subseteq V(F-r)\cup\{x_i, y_i, b\}$ and $V(P[a, x_i))$ has no neighbor in $V(B)$ (by the definition of x_i in Notation 3.9), it follows that P' is induced in $G-uu''$. Because of the path W, and since $P(a, a')$ is nonseparating in G_F, $G_F - V(P'(u, u''))$ is connected. So $P'(u, u'')$ is nonseparating in G_F. If $r \in \{u, u''\}$, then since $r \notin \{u, u'\}$, $r = u'' = a$ and r is not a cut vertex of $G_F - V(P(a, a'))$. Then, because of the path W, r is not a cut vertex of $G_F - V(P'(u, u''))$. Thus, P' is a feasible F-path. Since $B_P \cup W \subseteq B_{P'}$, P' is a B_P-augmenting path. Clearly, P' can be found in $O(|V(G)| + |E(G)|)$ time.

Subcase 1.2. B is nontrivial.

Let $S := N_G(B-b)-\{b, y_i\}$, and let G' be obtained from B by adding S and the edges of G from S to $V(B)-\{b\}$. Since $r \notin N_G(B-b)-\{b\}$, $r \notin S$. Since G is 4-connected, $|S| \geq 2$ and G' is $(3, S\cup\{b\})$-connected (if $y_i \notin N_G(B-b)$, then actually $|S| \geq 3$ and G' is $(4, S\cup\{b\})$-connected). Moreover, $G'-S = B$ is 2-connected. Thus, the hypotheses of Lemma 2.4 are satisfied with G', S, b as G, S, b, respectively. Then there exist $u, u' \in S$ and an induced u-u' path Q in G' such that $V(Q)\cap\{b\} = \emptyset$, $V(Q)\cap S = \{u, u'\}$, and $G' - (V(Q)\cup S)$ is connected. Moreover, such a path Q can be found in $O(|V(G')| + |E(G')|)$ time (and hence, in $O(|V(G)| + |E(G)|)$ time).

By the definition of x_i, y_i in Notation 3.9 and because $y_i \neq a'$, $\{u, u'\}\cap\{a'\} = \emptyset$. By symmetry we may assume that $u \neq x_i$. If $u' \neq x_i$, then let $P' := Q$; otherwise, let $P' := P[a, x_i]\cup Q$. Clearly, P' is a path with ends in $V(F)$ which is internally disjoint from $V(B_P)\cup V(F)$.

Next we show that P' is a B_P-augmenting path. Let u, u'' denote the ends of P'. By assumption, $N_G(P(a, a'))\cap V(F) \subseteq \{a, a'\}\cup\{r\}$ (Assumption 1), and P is induced in $G - aa'$. Then since Q is induced in G' and $P[a, x_i)$ has no neighbor in $V(B)$ (by the definition of x_i in Notation 3.9), it follows that P' is induced in $G-uu''$. Since $B - V(Q(u, u')) = G' - (V(Q)\cup S)$ is connected and because of the path W, $P'(u, u'')$ is nonseparating in G_F. If $r \in \{u, u''\}$, then since $r \notin S$, $r = u'' = a$, and r is not a cut vertex of $G_F - V(P(a, a'))$. Then, because of the path W, r is not a cut vertex of $G_F - V(P'(u, u''))$. Thus, P' is a feasible F-path. Since $B_P \cup W \subseteq B_{P'}$, P' is a B_P-augmenting path. Clearly, P' can be found in $O(|V(G)| + |E(G)|)$ time.

Case 2. y_i has no neighbor in $V(A_i) - (\{x_i, y_i\}\cup V(B-b))$ (and hence, $y_i \in N_G(B-b)$).

Subcase 2.1. B is trivial.

Let v denote the unique vertex in $V(B)-\{b\}$. Then $N_G(v) \subseteq V(F-r)\cup\{x_i, y_i, b\}$, and y_i is adjacent to v. Since G is 4-connected, v has at least four neighbors in $V(F-r)\cup\{x_i, y_i, b\}$, and hence, it has at least two neighbors in $V(F-r)\cup\{x_i\}$. Let $u, u' \in N_G(v)-\{b, y_i\}$, and assume that $u \neq x_i$. By the definition of x_i, y_i in Notation 3.9, one can see that $\{u, u'\}\cap\{a'\} = \emptyset$ (because $y_i \neq a'$) and $u \neq a$ (because $u \neq x_i$).

Suppose that there exists $n \in \{1,\dots,p\} - \{i,m\}$ such that V_n has a neighbor in $V(P[y_i,a'))$. Then there exists an r_m-r_n path W in $G_F - V(P(a,y_i))$ which is internally disjoint from $V(B_P)$ and intersects $P[y_i,a')$. If $u' \neq x_i$, then let $P' := (u,v,u')$; otherwise, let $P' := P[a,x_i] + \{u,v,uv,vx_i\}$. Then P' is a path with ends in $V(F)$ which is internally disjoint from $V(B_P) \cup V(F)$. Next we show that P' is a B_P-augmenting path. Let u,u'' denote the ends of P'. By assumption, $N_G(P(a,a')) \cap V(F) \subseteq \{a,a'\} \cup \{r\}$ (Assumption 1) and P is induced in $G - aa'$. Then, since $N_G(v) \subseteq V(F-r) \cup \{x_i,y_i,b\}$ and $V(P[a,x_i))$ has no neighbor in $V(B)$ (by the definition of x_i in Notation 3.9), it follows that P' is an induced path in $G - uu''$. Because of the path W and since $P(a,a')$ is nonseparating in G_F, $G_F - V(P'(u,u''))$ is connected, and so $P'(u,u'')$ is nonseparating in G_F. If $r \in \{u,u''\}$, then since $r \notin \{u,u'\}$, $r = u'' = a$, and r is not a cut vertex of $G_F - V(P(a,a'))$. Then, because of the path W, r is not a cut vertex of $G_F - V(P'(u,u''))$. Thus, P' is a feasible F-path. Since $B_P \cup W \subseteq B_{P'}$, P' is a B_P-augmenting path.

Thus, we may assume that there exists no $n \in \{1,\dots,p\} - \{i,m\}$ such that V_n has a neighbor in $V(P[y_i,a'))$. Since $|X_P| \geq 3$ and $V(P(x_i,y_i))$ has no neighbor in $(\bigcup_{j=1}^{p} V_j) - V_i$, we have that $x_i \neq a$, and there exists $n \in \{1,\dots,p\} - \{i,m\}$ such that V_n has a neighbor in $V(P(a,x_i])$. Furthermore, x_i has a neighbor in $V(A_i) - \{x_i,y_i,v\}$; otherwise, since y_i has no neighbor in $V(A_i) - (\{x_i,y_i\} \cup V(B-b))$, v would be a cut vertex of A_i. Therefore, there exists an r_i-r_n path W in $G_F - V(P'[y_i,a'))$ which is internally disjoint from $V(B_P)$ and intersects $P(a,x_i]$.

Let $P' := P[y_i,a'] + \{u,v,uv,vy_i\}$. Then P' is a path with ends in $V(F)$ which is internally disjoint from $V(B_P) \cup V(F)$. One can show that P' is an induced path in $G - ua'$, and because of the path W, $P'(u,a')$ is nonseparating in G_F. Since $u \neq x_i \neq r$, r is an end of P' only if $a' = r$. In this case, r is not a cut vertex of $G_F - V(P(a,a'))$, and because of the path W, r is not a cut vertex of $G_F - V(P'(u,a'))$. Thus, P' is a feasible F-path. Since $B_P \cup W \subseteq B_{P'}$, P' is a B_P-augmenting path. Note that in all above cases, such a path P' can be found in $O(|V(G)|+|E(G)|)$ time.

Subcase 2.2. B is nontrivial.

First, we define a graph G' from B. If y_i has at least two neighbors in $V(B)$, then let $S := N_G(B-b) - \{b,y_i\}$, let G' be obtained from B by adding $S \cup \{y_i\}$ and the edges of G from $S \cup \{y_i\}$ to $V(B) - \{b\}$, and let $y^* := y_i$. If y_i has exactly one neighbor in $V(B)$, then let y^* denote this vertex (note that $y^* \neq b$ because $y_i \in N_G(B-b)$ by assumption), let $S := N_G(B - \{b,y^*\}) - \{b,y^*\}$, and let G' be obtained from B by adding S and the edges of G from S to $V(B) - \{b,y^*\}$. Note that in either case $S \subseteq V(F-r) \cup \{x_i\}$. Moreover, $G' - S = B$ is 2-connected, and G' is $(4, S \cup \{b,y^*\})$-connected (because G is 4-connected). Thus, the hypotheses in Lemma 2.3 are satisfied with G', S, b, y^* as G, S, b, b', respectively. By Lemma 2.3 exactly one of the following holds:

(1) there exist $u,u' \in S$ and an induced u-u' path Q in G' such that $V(Q) \cap \{b,y^*\} = \emptyset$, $V(Q) \cap S = \{u,u'\}$, and $G' - (V(Q) \cup S)$ is connected; or

(2) $|S| = 2$, and the elements of S can be labeled as u,u' such that (G',u,b,u',y^*) is planar.

Moreover, one can in $O(|V(G')|+|E(G')|)$ time (and hence, in $O(|V(G)|+|E(G)|)$ time) find a path as in (1) or certify that (2) holds. Without loss of generality, we may assume that $u \neq x_i$.

Suppose (1) occurs. If $u' \neq x_i$, then let $P' := Q$; otherwise let $P' := P[a,x_i] \cup Q$. Then P' is a path with ends in $V(F)$ which is internally disjoint from $V(B_P) \cup V(F)$. Since y^* and b are in $G' - (V(Q) \cup S)$ which is connected, and because V_m has a neighbor in $V(P[y_i, a'))$, there exists an r_i-r_m path W in $G_F - V(P(a, y_i))$ which is internally disjoint from $V(B_P) \cup V(F)$ and intersects $P[y_i, a')$.

Next we show that P' is a B_P-augmenting path. Let u, u'' denote the ends of P'. Since Q is induced in G' and $N_G(P(a, a')) \cap V(F) \subseteq \{a, a'\} \cup \{r\}$, and because P is induced in $G - aa'$ and $P[a, x_i)$ has no neighbor in $V(B)$ (by the definition of x_i in Notation 3.9), one can see that P' is an induced path in $G - uu''$. Because of the path W, and since $P(a, a')$ is nonseparating in G_F, $P'(u, u'')$ is nonseparating in G_F. Since $r \notin S$, if $r \in \{u, u''\}$, then $r = u'' = a$, and r is not a cut vertex of $G_F - V(P(a, a'))$. Then, because of the path W, r is not a cut vertex of $G_F - V(P'(u, u''))$. Thus, P' is a feasible F-path. Since $B_P \cup W \subseteq B_{P'}$, P' is a B_P-augmenting path.

So we may assume (2) occurs. We consider two cases.

First, assume there exists $n \in \{1, \dots, p\} - \{i, m\}$ such that V_n has a neighbor in $V(P[y_i, a'))$. Then there exists an r_m-r_n path W in $G_F - V(P(a, y_i))$ which is internally disjoint from $V(B_P) \cup V(F)$ and intersects $P[y_i, a')$. By Lemma 2.6 (with G', u, u', b, y^* as G, a, a', b, b', respectively), there exists an induced u-u' path Q in G' such that $G' - V(Q)$ has exactly two components K and K' with $b \in V(K)$ and $y^* \in V(K')$. Moreover, such a path can be found in $O(|V(G')| + |E(G')|)$ time (and hence, in $O(|V(G)| + E(G)|)$ time). If $u' \neq x_i$, then let $P' := Q$; otherwise let $P' := P[a, x_i] \cup Q$. So P' is a path with ends in $V(F)$ which is internally disjoint from $V(B_P) \cup V(F)$.

Next we show that P' is a B_P-augmenting path. Let u, u'' denote the ends of P'. Since Q is induced in G' and $N_G(P(a, a')) \cap V(F) \subseteq \{a, a'\} \cup \{r\}$ (by Assumption 1), and because P is induced in $G - aa'$ and $P[a, x_i)$ has no neighbor in $V(B)$ (by the definition of x_i in Notation 3.9), one can see that P' is an induced path in $G - uu''$. Since $G' - V(Q)$ has exactly two components, one containing b and the other containing y^*, and because of the path W, it follows that $P'(u, u'')$ is nonseparating in G_F. If $r \in \{u, u''\}$, then $r = u'' = a$, and r is not a cut vertex of $G_F - V(P(a, a'))$. Then, because of the path W, r is not a cut vertex of $G - V(P'(u, u''))$. Thus, P' is a feasible F-path. Since $B_P \cup W \subseteq B_{P'}$, P' is a B_P-augmenting path.

Now assume that there exists no $n \in \{1, \dots, p\} - \{i, m\}$ such that V_n has a neighbor in $V(P[y_i, a'))$. Since $|X_P| \geq 3$ and $V(P(x_i, y_i))$ has no neighbor in $(\bigcup_{j=1}^{p} V_j) - V_i$, there exists $n \in \{1, \dots, p\} - \{i, m\}$ such that V_n has a neighbor in $V(P(a, x_i])$, and hence, $x_i \neq a$. Note that $G' \ncong K_{1,4}$ because B is nontrivial. By Lemma 2.7 (with G', u, y^*, u', b as G, a, a', b, b', respectively), there exists a nonseparating induced u-y^* path Q in G' such that $V(Q) \cap \{u', b\} = \emptyset$. Moreover, such a path can be found in $O(|V(G')| + |E(G')|)$ time (and hence, in $O(|V(G)| + |E(G)|)$ time). Note that either x_i has a neighbor in $V(A_i) - V(B - b)$ or x_i is in $G' - V(Q)$. Since V_n has a neighbor in $V(P(a, x_i])$, there exists an r_i-r_n path W in $G_F - V(P(x_i, a'))$ which is internally disjoint from $V(B_P) \cup V(F)$ and intersects $P(a, x_i]$. If $y^* = y_i$, then let $P' := Q \cup P[y_i, a']$; otherwise, let $P' := (Q \cup P[y_i, a']) + \{y_i, y_i y^*\}$. One can show that P' is an induced path in $G - ua'$, and because of the path W, $P'(u, a')$ is nonseparating in G_F. If $r \in \{u, a'\}$, then $a' = r$, r is not a cut vertex of $G_F - V(P'(u, a'))$ because of the path W, and because r is not a cut vertex of $G_F - V(P(a, a'))$. Thus, P' is a feasible F-path. Since $B_P \cup W \subseteq B_{P'}$, P' is a B_P-augmenting path. □

LEMMA 3.15. *Assume that $|X_P| \geq 3$, $|V(P)| \geq 4$, and for every $H_j \in \mathcal{H}$, $x_j \neq y_j$. Suppose that $H_i \in \mathcal{H}$ and $V(A_i) - \{b_i, x_i, y_i\}$ has a neighbor in $V(F - r) - \{a, a'\}$. Assume that $V(P(x_i, y_i))$ has a neighbor in $(\bigcup_{j=1}^{p} V_j) - V_i$. Then a B_P-augmenting path can be found in $O(|V(G)| + |E(G)|)$ time.*

Proof. Since $|X_P| \geq 3$ and $V(P(x_i, y_i))$ has a neighbor in $(\bigcup_{j=1}^{p} V_j) - V_i$, there exist $m, n \in \{1, \dots, p\} - \{i\}$ such that both V_m and V_n have a neighbor in $V(P(a, y_i))$, or both V_m and V_n have a neighbor in $V(P(x_i, a'))$.

By symmetry we may assume that both V_m and V_n have a neighbor in $V(P(x_i, a'))$. Therefore, there exists an r_m-r_n path W in $G_F - V(P(a, x_i])$ which is internally disjoint from $V(B_P) \cup V(F)$ and intersects $P(x_i, a')$.

Let D be the graph obtained from $A_i - \{x_i, y_i\}$ by adding a new vertex b' and new edges from b' to each $v \in V(P(x_i, y_i))$ such that v has a neighbor in some V_j, $j \in \{1, \dots, p\} - \{i\}$. Since $P(x_i, y_i) \subseteq A_i - \{x_i, y_i\}$, $N_D(b') \cup \{b'\}$ is contained in a block of D, and b' is not a cut vertex of D. Note also that if D is not connected, then D has exactly two components, one containing b_i and the other induced by $V(P(x_i, y_i)) \cup \{b'\}$, and the component containing b' is a block of D since every vertex in $V(P(x_i, y_i))$ has a neighbor in some V_j, $j \neq i$ (because $N_G(P(a, a')) \cap V(F) \subseteq \{a, a'\} \cup \{r\}$ by Assumption 1). We consider two cases.

Case 1. D is not a b_i-b' chain.

Then there exists an endblock B of D such that one of the following holds: (1) $b' \notin V(B)$, and if $b_i \in V(B)$, then b_i is a cut vertex of D, or (2) D has exactly two components and B is the component of D containing b_i (and hence, $V(B) \cap (V(P(x_i, y_i)) \cup \{b'\}) = \emptyset$ by the argument in the last paragraph). Note that such an endblock can be found in $O(|V(G)| + |E(G)|)$ time. If (1) holds, then let b denote the cut vertex of D contained in B. If (2) holds, then let $b := b_i$. Since $|X_P| \geq 3$ and B_P is a block of $G_F - V(P(a, a'))$, it follows from the definition of x_i, y_i in Notation 3.9 that $r \notin \{x_i, y_i\}$. Note that $N_D(b') \cap V(B - b) = \emptyset$ and $r \notin N_G(B - b)$.

Subcase 1.1. B is trivial.

Let v denote the only vertex in $V(B) - \{b\}$. Note that $N_G(v) \subseteq V(F - r) \cup \{x_i, y_i, b\}$. Since G is 4-connected and $N_D(b') \cap V(B - b) = \emptyset$, v has at least three neighbors in $V(F - r) \cup \{x_i, y_i\}$. Let u, u' be two distinct neighbors of v in $V(F - r) \cup \{x_i\}$. By symmetry, we may assume that $u \neq x_i$. If $u' \neq x_i$, then let $P' := (u, v, u')$. If $u' = x_i$, then let $P' := P[a, x_i] + \{u, v, uv, vx_i\}$. Then P' is a path with ends in $V(F)$ which is internally disjoint from $V(B_P) \cup V(F)$.

Next we show that P' is a B_P-augmenting path. Let u, u'' denote the ends of P'. By assumption, $N_G(P(a, a')) \cap V(F) \subseteq \{a, a'\} \cup \{r\}$, and P is induced in $G - aa'$. Since $N_G(v) \subseteq V(F - r) \cup \{x_i, y_i, b\}$ and $V(P[a, x_i))$ has no neighbor in $V(B)$ (by the definition of x_i in Notation 3.9), it follows that P' is induced in $G - uu''$. Because of the path W and since $P(a, a')$ is nonseparating in G_F, $P'(u, u'')$ is nonseparating in G_F. Moreover, if $r \in \{u, u''\}$, then $r = u'' = a$, and r is not a cut vertex of $G_F - V(P(a, a'))$. Then, because of the path W, r is not a cut vertex of $G_F - V(P'(u, a))$. Thus, P' is a feasible F-path. Since $B_P \cup W \subseteq B_{P'}$, P' is a B_P-augmenting path.

Subcase 1.2. B is nontrivial.

Let $S := N_G(B - b) - \{y_i, b\}$, and let G' be obtained from B by adding S and the edges of G from S to $V(B) - \{b\}$. Note that since $r \notin \{x_i, y_i\}$ and $r \notin N_G(B - b)$, $r \notin S$. Since G is 4-connected and y_i is the only possible neighbor of $V(B - b)$ not in $S \cup \{b\}$, G' is $(3, S \cup \{b\})$-connected. By Lemma 2.4 (with G', S, b as G, S, b, respectively) there exist $u, u' \in S$ and an induced u-u' path Q in G' such that $V(Q) \cap \{b\} = \emptyset$, $V(Q) \cap S = \{u, u'\}$, and $G' - (V(Q) \cup S)$ is connected. Moreover, such a path can be found in $O(|V(G')| + |E(G')|)$ time (and hence, in $O(|V(G)| + |E(G)|)$ time). Without loss of generality, we may assume that $u \neq x_i$. If $u' \neq x_i$, then let $P' := Q$; otherwise let $P' := P[a, x_i] \cup Q$. Then P' is a path with ends in $V(F)$ which is internally disjoint from $V(B_P) \cup V(F)$.

Next we prove that P' is a B_P-augmenting path. Let u, u'' denote the ends of P'. Note that Q is induced in $G - uu''$, $N_G(P(a, a')) \cap V(F) \subseteq \{a, a'\} \cup \{r\}$ (by Assumption 1), and P is induced in $G - aa'$. Then since $N_G(v) \subseteq V(F - r) \cup \{x_i, y_i, b\}$ and $V(P[a, x_i))$ has no neighbor in $V(B)$ (by the definition of x_i in Notation 3.9), it follows that P' is induced in $G - uu''$. Because of the path W and since $G' - (V(Q) \cup S)$ is connected, $P'(u, u'')$ is nonseparating in G_F. If $r \in \{u, u''\}$, then $r = u'' = a$, and r is not a cut vertex of $G_F - V(P(a, a'))$. Then, because of the path W, r is not a cut vertex of $G_F - V(P'(u, u''))$. Thus, P' is a feasible F-path. Since $B_P \cup W \subseteq B_{P'}$, P' is a B_P-augmenting path.

Case 2. D is a b_i-b' chain.

Let $D := w_0 B_1 w_1 \dots w_{l-1} B_l w_l$ where $w_0 := b_i$ and $w_l = b'$. Note that this block decomposition can be found in $O(|V(G)| + |E(G)|)$ time.

For each nontrivial block B_j with $1 \le j \le l - 1$, let $S_j := N_G(B_j - \{w_{j-1}, w_j\})$. If B_l is nontrivial, then let $S_l := N_G(B_l - \{w_{l-1}, w_l\}) - X_P$, namely, S_l contains all neighbors of $V(B_l - \{w_{l-1}, w_l\})$ except the neighbors of $N_D(b')$ contained in X_P. For each nontrivial block B_j with $1 \le j \le l$, let G_j be obtained from B_j by adding S_j and the edges of G from S_j to $V(B_j)$. Note that $N_D(b') \cup \{b'\} \subseteq B_l$, and for $1 \le j \le l-1$, $V(B_j) \cap (N_D(b') \cup \{b'\}) \subseteq \{w_{l-1}\}$. Hence, for $1 \le j \le l-1$, $S_j \subseteq V(F - r) \cup \{x_i, y_i\}$. Moreover, $r \notin \{x_i, y_i\}$ by Notation 3.9. Thus, $r \notin S_j$ for $1 \le j \le l - 1$. Also if B_l is nontrivial, then no vertex in $V(B_l - N_D(b'))$ is adjacent to r, and by the definition of S_l, $r \notin S_l$. First, we prove the following.

Claim. One can in $O(|V(G)| + |E(G)|)$ time either find a B_P-augmenting path or certify that the following statements hold:

(I) for each nontrivial block B_j with $1 \le j \le l - 1$, $|S_j| = 2$, $y_i \in S_j$, and if u denotes the vertex in $S_j - \{y_i\}$, then $(G_j, y_i, w_{j-1}, u, w_j)$ is planar;

(II) for each $1 \le j \le l - 2$ for which both B_j, B_{j+1} are trivial, $|N_G(w_j) - \{w_{j-1}, w_{j+1}\}| = 2$, and $y_i \in N_G(w_j)$; and

(III) if B_l is nontrivial, then $S_l \cap (V(F - r) - \{a, a'\}) = \emptyset$.

Proof of Claim. We will show that if one of (I)–(III) does not hold, then one can find in $O(|V(G)| + |E(G)|)$ time a B_P-augmenting path.

Proof of (I). Suppose that $j \in \{1, \dots, l - 1\}$ and B_j is nontrivial. Note that $G_j - S_j = B_j$ is 2-connected. Moreover, since G is 4-connected, G_j is $(4, S_j \cup \{w_{j-1}, w_j\})$-connected. Thus, the hypotheses of Lemma 2.3 are satisfied with G_j, S_j, w_{j-1}, w_j as G, S, b, b', respectively. By Lemma 2.3 exactly one of the following holds:

(1) there exist $u, u' \in S_j$ and an induced u-u' path Q such that $V(Q) \cap \{w_{j-1}, w_j\} = \emptyset$, $V(Q) \cap S_j = \{u, u'\}$, and $G_j - (V(Q) \cup S_j)$ is connected; or

(2) $|S_j| = 2$, and the elements of S_j can be labeled as u, u' such that $(G_j, u, w_{j-1}, u', w_j)$ is planar.

Moreover, one can in $O(|V(G_j)| + |E(G_j)|)$ time (and hence, in $O(|V(G)| + |E(G)|)$ time) find a path as in (1) or certify that (2) holds.

Suppose that (1) holds. Define P' as follows.

(a) if $\{u, u'\} \cap \{x_i, y_i\} = \emptyset$, then let $P' := Q$;

(b) if $\{u, u'\} = \{x_i, y_i\}$, then let $P' := (P - V(P(x_i, y_i))) \cup Q$;

(c) if $\{u, u'\} \cap \{x_i, y_i\} = \{x_i\}$, then let $P' := P[a, x_i] \cup Q$; and

(d) if $\{u, u'\} \cap \{x_i, y_i\} = \{y_i\}$, then let $P' := P[y_i, a'] \cup Q$.

We claim that P' is a path with ends in $V(F)$ which is internally disjoint from $V(B_P) \cup V(F)$. If (a) or (b) occurs, then clearly P' is a path as claimed. Suppose (c) occurs, that is, $\{u, u'\} \cap \{x_i, y_i\} = \{x_i\}$. If $a \notin \{u, u'\}$, then clearly P' is a path as claimed;

if $a \in \{u, u'\}$, then by the definition of x_i in Notation 3.9, $x_i = a$, and hence, P' is a path as claimed. Similarly, if (d) occurs, then P' is a path as claimed.

Next we show that P' is a B_P-augmenting path. Let u_1, u_2 denote the ends of P'. Since Q is induced in G_j and $N_G(P(a, a')) \cap V(F) \subseteq \{a, a\} \cup \{r\}$ (by Assumption 1), and because P is induced in $G - aa'$ and $P[a, x_i) \cup P(y_i, a']$ has no neighbor in $V(B_j)$ (by the definition of x_i and y_i in Notation 3.9), one can show that P' is an induced path in $G - u_1u_2$. Since $G_j - (V(Q) \cup S_j)$ is connected, it is easy to see that $P'(u_1, u_2)$ is nonseparating in G_F. If $r \in \{u_1, u_2\}$, then since $r \notin \{u, u'\} \subseteq S_j$, (b), (c), or (d) occurs and either $r = a$ or $r = a'$. In this case, r is not a cut vertex of $G_F - V(P(a, a'))$, and since $|X_P| \geq 3$, r is not a cut vertex of $G_F - V(P'(u_1, u_2))$. Thus, P' is a feasible F-path. Since there exists a w_{j-1}-w_j path in $G_j - (V(Q) \cup S_j)$, there exists an r_i-b' path in $D - V(P'(u_1, u_2))$. By the definition of b', the vertex adjacent to b' in this path has a neighbor in V_t for some $t \in \{1, \dots, p\} - \{i\}$. Hence, B_P is properly contained in a block of $G_F - V(P'(u_1, u_2))$, and therefore, P' is a B_P-augmenting path.

So assume that (2) holds. If $y_i \in \{u, u'\}$, then (I) holds, so we may assume that $y_i \notin \{u, u'\}$. By Lemma 2.6 with G_j, u, u', w_{j-1}, w_j as G, a, a', b, b', respectively, one can find in $O(|V(G_j)| + |E(G_j)|)$ time an induced u-u' path Q such that $G_j - V(Q)$ has exactly two components K and K' with $w_{j-1} \in V(K)$ and $w_j \in V(K)$. Without loss of generality, we may assume that $u \neq x_i$. If $u' \neq x_i$, then let $P' := Q$; otherwise, let $P' := P[a, x_i] \cup Q$. Clearly, P' is a path with ends in $V(F)$ which is internally disjoint from $V(B_P) \cup V(F)$.

Next we show that P' is a B_P-augmenting path. Let u, u'' denote the ends of P'. Since Q is induced in G_j and $N_G(P(a, a')) \cap V(F) \subseteq \{a, a'\} \cup \{r\}$, and because P is an induced path in $G - aa'$ and $P[a, x_i)$ has no neighbor in $V(B_j) - \{w_{j-1}, w_j\}$ (by the definition of x_i in Notation 3.9), one can show that P' is an induced path in $G - uu''$. Since $G_j - V(Q)$ has exactly two components, one containing w_{j-1} and the other containing w_j, and because of the path W, it follows that $G - V(P'(u, u''))$ is connected, so P' is nonseparating in G_F. If $r \in \{u, u''\}$, then $r = u'' = a$, and r is not a cut vertex of $G_F - V(P(a, a'))$. Then, because of the path W, r is not a cut vertex of $G_F - V(P'(u, u'')$. Thus, P' is a feasible F-path. Since $B_P \cup W \subseteq B_{P'}$, P' is a B_P-augmenting path.

Proof of (II). Suppose that for some $j \in \{1, \dots, l-1\}$ both B_j and B_{j+1} are trivial. If $y_i \in N_G(w_j)$ and $|N_G(w_j) - \{w_{j-1}, w_j\}| = 2$, then (II) holds, so we may assume that $y_i \notin N_G(w_j)$ or $|N_G(w_j) - \{w_{j-1}, w_j\}| \neq 2$. Therefore, $|N_G(w_j) - \{w_{j-1}, w_j, y_i\}| \geq 2$. Let u, u' be distinct vertices in $N_G(w_j) - \{w_{j-1}, w_j, y_i\}$. Note that $r \notin \{u, u'\}$ because B_P is a block of $G_F - V(P(a, a'))$. Without loss of generality we may assume that $u \neq x_i$. If $u' \neq x_i$, then let $P' := (u, w_j, u')$. If $u' = x_i$, then let $P' := P[a, x_i] + \{u, w_j, w_jx_i, uw_j\}$. By the definition of x_i, y_i in Notation 3.9, $u \neq a$ when $u' = x_i$. So P' is a path with ends in $V(F)$ which is internally disjoint from $V(B_P) \cup V(F)$. Note that $V(P') \cap V(P(x_i, a')) = \emptyset$.

Next we show that P' is a B_P-augmenting path. Let u, u'' denote the ends of P. Since P is an induced path in $G - aa'$, and because w_j has no neighbor in $P[a, x_i)$ (by the definition of x_i in Notation 3.9), one can see that P' is induced in $G - uu''$. Because of the path W and since $P(a, a')$ is nonseparating in G_F, P' is nonseparating in G_F. If $r \in \{u, u''\}$, then $r = u'' = a$, and r is not a cut vertex of $G_F - V(P(a, a'))$. Then, because of the path W, r is not a cut vertex of $G_F - V(P'(u, u''))$. Thus, P' is a feasible F-path. Since $B_P \cup W \subseteq B_{P'}$, P' is a B_P-augmenting path.

Proof of (III). Suppose B_l is nontrivial. If $S_l \cap (V(F-r) - \{a, a'\}) = \emptyset$, then (III) holds, so we may assume that $S_l \cap (V(F-r) - \{a, a'\}) \neq \emptyset$. We want to apply Lemma 2.3 to find a B_P-augmenting path, so we need to show that $G_l, S_l, w_{l-1}, w_l = b'$ (as G, S, b, b', respectively) satisfy the hypotheses in the statement of Lemma 2.3. Clearly, $G_l - S_l = B_l$ is 2-connected and by definition, every vertex in S_l has a neighbor in $V(B_l) - \{w_{l-1}, w_l\}$. Since P is an induced path in $G - aa'$ and G is 4-connected, $G_l - b' \subseteq G$ is $(4, S_l \cup \{w_{l-1}\} \cup N_D(b'))$-connected. Hence, G_l is $(3, S_l \cup \{w_{l-1}, b'\})$-connected. Recall that $r \notin S_l$ (see the definition of S_l), $V(P(x_i, y_i))$ has no neighbor in $V(F-r) - \{a, a'\}$ unless $x_i = a$ or $y_i = a'$ (by Assumption 1), and $A_i - V(P_i)$ is connected. Thus, since $S_l \cap (V(F-r) - \{a, a'\}) \neq \emptyset$, $V(B_l) - (\{w_{l-1}, w_l\} \cup V(P(x_i, y_i))) \neq \emptyset$. This implies that $V(P(x_i, y_i)) \subseteq V(B_l) - \{w_{l-1}, w_l\}$ (and hence, $x_i, y_i \in S_l$); otherwise, $w_{l-1} \in V(P(x_i, y_i))$, contradicting the fact that $A_i - V(P_i)$ is connected. Thus, $|S_l| \geq 3$, and there exists a component K of $G_l - (S_l \cup \{w_{l-1}, w_l\}) = B_l - \{w_{l-1}, w_l\}$ which contains $V(P(x_i, y_i))$. Note that K has at least two neighbors in S_l, namely, x_i, y_i. Thus, the hypotheses of Lemma 2.3 are satisfied with G_l, S_l, w_{l-1}, w_l as G, S, b, b', respectively.

Therefore, by Lemma 2.3 there exist $u, u' \in S_l$ and an induced path Q in G_l such that $V(Q) \cap \{w_{l-1}, w_l\} = \emptyset$, $V(Q) \cap S_l = \{u, u'\}$, and $G_l - (V(Q) \cup S_l)$ is connected. Define P' as follows:

(a) if $\{u, u'\} \cap \{x_i, y_i\} = \emptyset$, then let $P' := Q$;
(b) if $\{u, u'\} = \{x_i, y_i\}$, then let $P' := (P - V(P(x_i, y_i))) \cup Q$;
(c) if $\{u, u'\} \cap \{x_i, y_i\} = \{x_i\}$, then let $P' := P[a, x_i] \cup Q$; and
(d) if $\{u, u'\} \cap \{x_i, y_i\} = \{y_i\}$, then let $P' := P[y_i, a'] \cup Q$.

We claim that P' is a path with ends in $V(F)$ which is internally disjoint from $V(B_P) \cup V(F)$. Clearly, this is true if (a) or (b) occurs. Suppose (c) occurs, that is, $\{u, u'\} \cap \{x_i, y_i\} = \{x_i\}$. If $a \notin \{u, u'\}$, then P' is a path as claimed. If $a \in \{u, u'\}$, then by the definition of x_i in Notation 3.9, $a = x_i$. Again, P' is a path as claimed. Similarly, if (d) occurs, then P' is a path as claimed.

Next we show that P' is a B_P-augmenting path. Let u_1, u_2 denote the ends of P'. Since Q is induced in G_l and $N_G(P(a, a')) \cap V(F) \subseteq \{a, a\} \cup \{r\}$, and because P is induced in $G - aa'$ and $P[a, x_i) \cup P(y_i, a']$ has no neighbor in B_l (by the definition of x_i and y_i in Notation 3.9), one can see that P' is an induced path in $G - u_1u_2$. Since $G_l - (V(Q) \cup S_l)$ is connected and $V(P(x_i, y_i))$ has a neighbor in $(\bigcup_{j=1}^{p} V_j) - V_i$, it is easy to see that $P'(u_1, u_2)$ is nonseparating in G_F. If $r \in \{u_1, u_2\}$, then since $r \notin S_l$, (b), (c), or (d) occurs, and either $r = a$ or $r = a'$. In this case, r is not a cut vertex of $G_F - V(P(a, a'))$, and since $|X_P| \geq 3$, r is not a cut vertex of $G_F - V(P'(u_1, u_2))$. Thus, P' is a feasible F-path. Moreover, since there exists a w_{l-1}-w_l path W' in $G_l - (V(Q) \cup S_l)$, there exists an r_i-b' path W'' in $D - V(P'(u_1, u_2))$. By the definition of b', the vertex adjacent to b' in W'' has a neighbor in V_t for some $t \in \{1, \ldots, p\} - \{i\}$. Hence, B_P is properly contained in a block of $G_F - V(P'(u_1, u_2))$, and therefore, P' is a B_P-augmenting path.

This concludes the proof of the claim. $\square$

By the above claim, we may assume that (I), (II), and (III) hold. Therefore, by (III) and since $V(A_i) - \{b_i, x_i, y_i\}$ has a neighbor in $V(F-r) - \{a, a'\}$, we have $l \geq 2$. We consider three subcases.

Subcase 2.1. x_i has at least two neighbors in $V(B_l)$.

Thus, B_l is nontrivial (because x_i is not adjacent to b' in D). We claim that $P(x_i, y_i) \subseteq B_l - w_{l-1}$. Suppose for a contradiction that $P(x_i, y_i) \not\subseteq B_l - w_{l-1}$. Then $w_{l-1} \in V(P(x_i, y_i))$. Since $G_F - V(P(a, a'))$ is connected, $B_l - b' \subseteq P(x_i, y_i)$. But then x_i has at most one neighbor in $V(B_l)$ because P is an induced path in $G_F - aa'$, a contradiction. Therefore, $P(x_i, y_i) \subseteq B_l - w_{l-1}$.

Since $V(A_i) - \{b_i, x_i, y_i\}$ has a neighbor in $V(F - r) - \{a, a'\}$ and $S_l \cap (V(F - r) - \{a, a'\}) = \emptyset$ by (III), there exists $q \in \{1, \ldots, l-1\}$ such that $V(B_q - w_{q-1})$ has a neighbor in $V(F - r) - \{a, a'\}$. Choose q to be maximum with this property, and let u be a neighbor of $V(B_q - w_{q-1})$ in $V(F - r) - \{a, a'\}$.

Next we define a u-w_q path Q_q in G_q. If B_q is trivial or u is adjacent to w_q, then let Q_q be the path induced by the edge uw_q. Otherwise, B_q is nontrivial, $S_q = \{u, y_i\}$, and $(G_q, y_i, w_{q-1}, u, w_q)$ is planar (by (I)). By Lemma 2.7 (with $G_q, u, w_q, y_i, w_{q-1}$ as G, a, a', b, b', respectively), there exists a nonseparating induced u-w_q path Q_q in G_q such that $V(Q_q) \cap \{y_i, w_{q-1}\} = \emptyset$. Moreover, such a path can be found in $O(|V(G_q)| + |E(G_q)|)$ time.

By the maximality of q, for $q + 1 \leq j \leq l - 1$, the following holds: If B_j is nontrivial, then $S_j = \{x_i, y_i\}$ and $(G_j, y_i, w_{j-1}, x_i, w_j)$ is planar (by (I)), and if B_j and B_{j+1} are trivial, then $N_G(w_j) - \{w_{j-1}, w_{j+1}\} = \{x_i, y_i\}$ (by (II)). Note also that $x_i \in S_l$ because $P(x_i, y_i) \subseteq B_l - w_{l-1}$.

Choose the minimum $t \in \{q + 1, \ldots, l\}$ such that $x_i \in N_G(B_t - w_t)$. Thus, by the choice of q and t, B_j is trivial for every $j \in \{q + 1, \ldots, t - 1\}$. For each $j \in \{q + 1, \ldots, t - 1\}$, let Z_j denote the path induced by the edge $w_{j-1}w_j$.

If B_t is trivial, then let Q_t denote the path induced by the edge $w_{t-1}x_i$. If B_t is nontrivial, then we define a path Q_t according to the following two cases.

- $t < l$. Then $S_t = \{x_i, y_i\}$, and $(G_t, w_{t-1}, x_i, w_t, y_i)$ is planar. By Lemma 2.7 with $G_t, w_{t-1}, x_i, w_t, y_i$ as G, a, a', b, b', respectively, there exists a nonseparating induced w_{t-1}-x_i path Q_t in G_t such that $V(Q_t) \cap \{w_t, y_i\} = \emptyset$. Moreover, such a path can be found in $O(|V(G_t)| + |E(G_t)|)$ time.
- $t = l$. Since P is induced in $G - aa'$ and x_i has at least two neighbors in $V(B_l)$, x_i has a neighbor in $V(B_l) - V(P(x_i, y_i))$. Moreover, $B_l - V(P(x_i, y_i))$ is connected because $A_i - V(P_i)$ is connected, and hence, there exists a w_{l-1}-x_i path Q' in $B_l - V(P(x_i, y_i))$. Let $G' := G_l - b'$, and let $S' := N_D(b') \cup S_l \cup \{w_{l-1}\}$. Then G' is $(4, S')$-connected, and $S' - \{w_{l-1}, x_i\}$ is contained in a component U of $G' - V(Q')$. By Lemma 2.1 (with G', S', w_{l-1}, x_i, U as G, S, a, a', U, respectively) there exists a nonseparating induced w_{l-1}-x_i path Q_l in G' such that $V(Q_l) \cap V(U) = \emptyset$ (and hence, $V(Q_l) \cap V(P(x_i, y_i)) = \emptyset$). Moreover, such a path can be found in $O(|V(G')| + |E(G')|)$ time (and hence, in $O(|V(G)| + |E(G)|)$ time).

Let $P' := Q_q \cup Z_{q+1} \cup \cdots \cup Z_{t-1} \cup Q_t \cup P[a, x_i]$. Then P' is a u-a path in G such that $V(P') \cap V(F) = \{u, a\}$. Moreover, it is not hard to see that such a path can be found in $O(|V(G)| + |E(G)|)$ time.

Next we show that P' is a B_P-augmenting path. It is not hard to see that P' is an induced path in $G - ua$. Because of the path W and since $P(a, a')$ is nonseparating in G_F, $P'(u, a)$ is nonseparating in G_F. If $a = r$, then r is not a cut vertex of $G_F - V(P(a, a'))$, and because of the path W, r is not a cut vertex of $G_F - V(P'(u, a))$. Thus, P' is a feasible F-path. Moreover, since $V(P') \cap V(P(x_i, a']) = \emptyset$, $B_P \cup W \subseteq B_{P'}$. Therefore, P' is a B_P-augmenting path.

Subcase 2.2. x_i has at most one neighbor in $V(B_l)$, and x_i has a neighbor in $V(A_i) - (V(P(x_i, y_i)) \cup \{b_i\})$.

Then since A_i is 2-connected, x_i has a neighbor in $V(D) - (V(B_l) \cup \{b_i\})$. Therefore, since $V(A_i) - \{b_i, x_i, y_i\}$ has a neighbor in $V(F - r) - \{a, a'\}$ and by (I), (II), and (III), there exist $u \in V(F-r) - \{a, a'\}$ and $q, t \in \{1, \ldots, l-1\}$ with $q \leq t$ such that one of the following holds:

(a) $u \in N_G(B_q - w_{q-1})$, and $x_i \in N_G(B_t - w_t)$; or

(b) $x_i \in N_G(B_q - w_{q-1})$, and $u \in N_G(B_t - w_t)$.

Choose q, t so that $t - q$ is minimum and (a) or (b) holds. Note that $q < t$ because in (I) we must have $y_j \in S_j$ and in (II) we must have $y_j \in N_G(w_j)$.

We may assume that (a) holds because the other case is symmetric.

By the minimality of $t-q$ and by (I), B_j is trivial for every $j \in \{q+1, \ldots, t-1\}$. Using (II), one can also show that $t - q \leq 2$. For $q+1 \leq j \leq t-1$, let Z_j denote the path induced by the edge $w_{j-1}w_j$.

If B_q is trivial, then let Q_q be the path induced by the edge uw_q. Otherwise (by (I)) B_q is nontrivial, $S_q = \{u, y_i\}$, and $(G_q, y_i, w_{q-1}, u, w_q)$ is planar . By Lemma 2.7 (with $G_q, u, w_q, y_i, w_{q-1}$ as G, a, a', b, b', respectively), there exists a nonseparating induced u-w_q path Q_q in G_q such that $V(Q_q) \cap \{y_i, w_{q-1}\} = \emptyset$. Moreover, such a path can be found in $O(|V(G_q)| + |E(G_q)|)$ time.

Similarly, if B_t is trivial, then let Q_t be the path induced by the edge x_iw_{t-1}. Otherwise (by (I)) B_t is nontrivial, $S_t = \{x_i, y_i\}$, and $(G_t, y_i, w_{t-1}, x_i, w_t)$ is planar. By Lemma 2.7 (with $G_t, x_i, w_{t-1}, y_i, w_t$ as G, a, a', b, b', respectively) there exists a nonseparating induced x_i-w_{t-1} path Q_t in G_t such that $V(Q_t) \cap \{y_i, w_t\} = \emptyset$. Moreover, such a path can be found in $O(|V(G_t)| + |E(G_t)|)$ time.

Let $P' := Q_q \cup Z_{q+1} \cup \cdots \cup Z_{t-1} \cup Q_t \cup P[a, x_i]$. Then P' is a u-a path which is internally disjoint from $V(B_P) \cup V(F)$. Moreover, it is not hard to see that such a path can be found in $O(|V(G)| + |E(G)|)$ time.

Next we show that P' is a B_P-augmenting path. Since Q_q, Q_t are nonseparating and induced in G_q, G_t, respectively, it is not hard to see that P' is an induced path in $G - ua$. Because of the path W and since $P(a, a')$ is nonseparating in G_F, $P'(u, a)$ is non-separating in G_F. If $a = r$, then r is not a cut vertex of $G_F - V(P(a, a'))$, and because of the path W, r is not a cut vertex of $G_F - V(P'(u, a))$. Thus, P' is a feasible F-path. Since $V(P') \cap V(P_i - x_i) = \emptyset$, $B_P \cup W \subseteq B_{P'}$. Therefore, P' is a B_P-augmenting path.

Subcase 2.3. x_i has at most one neighbor in $V(B_l)$, and x_i has no neighbor in $V(A_i) - (V(P(x_i, y_i)) \cup \{b_i\})$.

In this case, since A_i is 2-connected, b_i is the only neighbor of x_i in A_i not contained in $V(P(x_i, y_i))$. We consider two cases according to whether $x_i = a$ or $x_i \neq a$.

(A) $x_i = a$.

Then by the definition of x_i in Notation 3.9, $b_i \neq r_i$. Since $V(A_i) - \{b_i, x_i, y_i\}$ has a neighbor in $V(F-r) - \{a, a'\}$ and (III) holds, there exists some $q \in \{1, \ldots, l-1\}$ such that $V(B_q - w_{q-1})$ has a neighbor in $V(F-r) - \{a, a'\}$. Choose q to be minimum with this property.

Therefore, since b_i is the only neighbor of x_i in A_i not contained in $V(P_i)$ and (I) holds, B_j is trivial for every $j \in \{1, \ldots, q-1\}$. Using (II), one can show that either $q = 1$ or $q = 2$. For each $j \in \{1, \ldots, q-1\}$ let Z_j be the path induced by the edge $w_{j-1}w_j$.

If B_q is trivial (in this case $q = 1$), then, by the choice of q, w_q has a neighbor u in $V(F-r)$, and let $Q_q := (w_{q-1}, w_q, u)$. If B_q is nontrivial, then by (I) $S_q = \{u, y_i\}$ for some $u \in V(F-r) - \{a, a'\}$, and $(G_q, y_i, w_{q-1}, u, w_q)$ is planar. Note that $u \neq a$ because x_i has no neighbor in $V(A_i) - (V(P_i) \cup \{b_i\})$. By Lemma 2.7 (with $G_q, u, w_{q-1}, y_i, w_q$ as G, a, a', b, b', respectively), there exists a nonseparating induced

u-w_{q-1} path Q_q in G_q such that $V(Q_q) \cap \{y_i, w_q\} = \emptyset$. Moreover, such a path can be found in $O(|V(G_q)| + |E(G_q)|)$ time.

Let $P' := (Z_1 \cup \cdots \cup Z_{q-1} \cup Q_q) + \{x_i, x_i b_i\}$. Then P' is a u-a path which is internally disjoint from $V(B_P) \cup V(F)$. Moreover, it is not hard to see that such a path can be found in $O(|V(G)| + |E(G)|)$ time.

Next we show that P' is a B_P-augmenting path. It is not hard to see that P' is an induced path in $G - ua$. Because of the path W and since $P(a, a')$ is nonseparating in G_F and Q_q is nonseparating in G_q, $P'(u, a)$ is nonseparating in G_F. If $a = r$, then r is not a cut vertex of $G_F - V(P(a, a'))$, and because of the path W, r is not a cut vertex of $G_F - V(P'(u, a))$. Thus, P' is a feasible F-path. Since $B_P \cup W \subseteq B_{P'}$, P' is a B_P-augmenting path.

(B) $x_i \neq a$.

In this case, it is possible that $b_i = r_i$. Note that x_i has degree at least four in G (because G is 4-connected), P is induced in $G - aa'$, and x_i has no neighbor in $V(A_i) - (V(P_i) \cup \{b_i\})$ (by assumption in this subcase). So x_i has a neighbor in $(\bigcup_{j=1}^{p} V_j) - V_i$. Let $t \in \{1, \ldots, p\} - \{i\}$ such that x_i has a neighbor in V_t.

Suppose that for some $j \in \{1, \ldots, l-1\}$, B_j is nontrivial. Then by (I) and by our assumption that x_i has no neighbor in $V(A_i) - (V(P(x_i, y_i)) \cup \{b_i\})$, $S_j = \{u, y_i\}$ for some $u \in V(F - r)$, and $(G_j, y_i, w_{j-1}, u, w_j)$ is planar. Note that $u \neq a'$ by the definition of y_i in Notation 3.9 and because $u \neq y_i$. Also $u \neq a$ because $x_i \neq a$. By Lemma 2.6 (with $G_j, y_i, u, w_{j-1}, w_j$ as G, a, a', b, b', respectively), there exists a nonseparating induced u-y_i path Q in G_j such that $G_j - V(Q)$ has exactly two components K and K' with $w_{j-1} \in V(K)$ and $w_j \in V(K')$. Moreover, such a path can be found in $O(|V(G_j)| + |E(G_j)|)$ time (and hence, in $O(|V(G)| + |E(G)|)$ time). Let $P' := Q \cup P[y_i, a']$. Then P' is a u-a' path in G such that $V(P') \cap V(F) = \{u, a'\}$. Moreover, it is not hard to see that such a path can be found in $O(|V(G)| + |E(G)|)$ time.

Next we show that P' is a B_P-augmenting path. Since Q is induced in G_j and $N_G(P(a, a')) \cap V(F) \subseteq \{a, a'\} \cup \{r\}$ (by Assumption 1), and because P is an induced path in $G - aa'$ and $P((y_i, a'])$ has no neighbor in $V(B_j)$ (by the definition of y_i in Notation 3.9), one can see that P' is an induced path in $G - ua'$. Since $G_j - V(Q)$ has exactly two components, one containing w_{j-1} and the other containing w_j, and because x_i has a neighbor in V_t, it is not hard to show that P' is nonseparating in G_F. If $r \in \{u, a'\}$, then $r = a'$ and r is not a cut vertex of $G_F - V(P(a, a'))$. In this case, because x_i has a neighbor in V_t, r is not a cut vertex of $G_F - V(P'(u, a'))$. Thus, P' is a feasible F-path. Moreover, since b_i is adjacent to x_i and x_i has a neighbor in V_t, it follows that P' is a B_P-augmenting path.

Thus, we may assume that B_j is trivial for every $j \in \{1, \ldots, l-1\}$. If $l \geq 3$, then B_1 and B_2 are trivial, and by (II), $N_G(w_1) - \{w_0, w_2\} = \{u, y_i\}$ for some $u \in V(F - r)$. Note that $u \notin \{a, a'\}$ because $x_i \neq a$ and $y_i \neq u$. By an argument similar to the above paragraph, one can show that $P' := (u, w_1, y_i) \cup P[y_i, a']$ is a B_P-augmenting path.

So we may assume that $l = 2$ and B_1 is trivial. This implies that $V(P(x_i, y_i)) \subseteq V(B_2)$. Hence, B_2 is nontrivial, so $S_2 = \{x_i, y_i\}$ (by (III)). Since $V(A_i) - \{b_i, x_i, y_i\}$ has a neighbor in $V(F - r) - \{a, a'\}$ (by assumption in this lemma) and (III) holds, w_1 is adjacent to some $u \in V(F - r) - \{a, a'\}$. Let x', y' denote the vertices in $N_D(b')$ (see Notation 3.9) which are the lowest and the highest in P, respectively. Since B_2 is 2-connected, $V(B_2) - (V(P(x_i, y_i)) \cup \{b'\})$ has a neighbor in $V(P(x', y_i))$. Since $B_2 - (V(P(x_i, y_i)) \cup \{b'\})$ is connected (because $A_i - V(P_i)$ is connected), there exists a w_1-y_i path Q' in G_2 such that x_i and b' are contained in a component U of $G_2 - V(Q')$.

Moreover, recall that G_2 is $(3, \{w_1, x_i, y_i, b'\})$-connected. Thus, the hypotheses of Lemma 2.1 are satisfied with $G_2, \{w_1, x_i, y_i, b'\}, w_1, y_i, U$ as G, S, a, a', U, respectively. By Lemma 2.1 there exists a nonseparating induced w_1-y_i path Q in G_2 such that $V(Q) \cap V(U) = \emptyset$ (and hence, $V(Q) \cap \{x_i, b'\} = \emptyset$). Moreover, such a path can be found in $O(|V(G_2)| + |E(G_2)|)$ time (and hence, in $O(|V(G)| + |E(G)|)$ time). Let $P' := (P[y_i, a'] \cup Q) + \{u, uw_1\}$. Then P' is a u-a' path in G such that $V(P') \cap V(F) = \{u, a'\}$. Moreover, it is not hard to see that such a path can be found in $O(|V(G)| + |E(G)|)$ time.

We conclude the proof by showing that P' is a B_P-augmenting path. Since Q is induced in G_2 and $N_G(P(a, a')) \cap V(F) \subseteq \{a, a'\} \cup \{r\}$ (by Assumption 1), and because P is an induced path in $G - aa'$ and $P((y_i, a'])$ has no neighbor in $V(B_2)$ (by the definition of y_i in Notation 3.9), one can see that P' is an induced path in $G - ua'$. Since $G_2 - V(Q)$ is connected, and because x_i has a neighbor in V_t, it is not hard to see that P' is nonseparating in G_F. If $r \in \{u, a'\}$, then $r = a'$, and r is not a cut vertex of $G_F - V(P(a, a'))$. In this case, because x_i has a neighbor in V_t, r is not a cut vertex of $G_F - V(P'(u, a'))$. Thus, P' is a feasible F-path. Moreover, since b_i is adjacent to x_i and x_i has a neighbor in V_t, it follows that P' is a B_P-augmenting path. $\square$

We are now ready to prove the main result of this section, which implies Theorem 3.2. Consider Algorithm 1.

Theorem 3.16. *Algorithm* 1 *is correct and runs in* $O(|V(G)||E(G)|)$ *time.*

Proof. First, we will prove the correctness of the algorithm.

At the start of each iteration of the main loop, P is a feasible a-a' F-path, and B_P is a nontrivial block of $G_F := G - V(F - r)$ containing r. As the algorithm progresses, $|V(B_P)|$ increases.

If $G_F - V(P(a, a'))$ is 2-connected, then the algorithm stops at line 5. Since P is an induced path in $G_F - aa'$, $H := P$ is either an elementary F-chain or an up a-a' F-chain whose blocks are all trivial. Moreover, $G_F - I(H) = G_F - V(P(a, a'))$ and $G[V(F) \cup I(H)] = F \cup P$ are 2-connected.

If for every B_P-bridge B of $G_F - V(P(a, a'))$, $N_G(B - r_B) \subseteq V(P)$, then by Lemma 3.6 the a-a' F-chain H in line 8 exists, and $G_F - I(H)$ and $G[V(F) \cup I(H)]$ are 2-connected. Thus, if the algorithm stops at line 9, it returns a correct answer.

If $|X_P| = 2$, then by Lemma 3.7 either the subgraph H defined in line 12 is a down F-chain or there exists a B_P-augmenting path. Thus, if the algorithm stops at line 14, then H is a down F-chain and $G_F - I(H) = B_P$ and $G[V(F) \cup I(H)]$ are 2-connected. Otherwise, the algorithm increases B_P by executing lines 16 and 17.

In line 19, if $|V(P)| = 3$ (and hence, $|X_P| \geq 3$), then $G_F - V(P(a, a'))$ is not 2-connected; for otherwise, Algorithm 1 would have stopped at line 5. By Lemma 3.8 a B_P-augmenting path exists, and the algorithm increases B_P.

Suppose then that $|X_P| \geq 3$ and $|V(P)| \geq 4$. Let $H_i \in \mathcal{H}$ be adjacent to F (see Notation 3.9). If $x_i = y_i$, then by Lemma 3.12 the B_P-augmenting path in line 24 exists, and the algorithm increases B_P. If $x_i \neq y_i$, then by Lemmas 3.12, 3.13, 3.14, and 3.15 either the subgraph H defined in line 26 is a triangle chain, or there exists a B_P-augmenting path. Thus, if the algorithm stops at line 28, then H is a triangle F-chain such that $G_F - I(H) = B_P$ and $G[V(F) \cup I(H)]$ are 2-connected. Otherwise, the algorithm increases B_P by executing lines 30 and 31.

Since $|V(B_P)|$ increases at each iteration, the main loop at line 1 eventually stops and a good F-chain in G is returned. Hence, Algorithm 1 is correct.

Algorithm 1. Internal Chain.

```
Require: $G, r, F, a, a', P, B_P$ satisfying the hypotheses of Theorem 3.2.
Return: A good $F$-chain $H$ in $G$ such that $G_F - I(H)$ and $G[V(F) \cup I(H)]$ are
    2-connected.
 1: loop
 2:   Apply Lemma 3.4 to $P$, and let $P$ denote the resulting path;
 3:   Let $a, a'$ denote the ends of $P$;
 4:   if $G_F - V(P(a, a'))$ is 2-connected then
 5:     Return $H := P$ and stop;
 6:   Compute $X_P$ (as defined in Notation 3.5);
 7:   if for every $B_P$-bridge $B$ of $G_F - V(P(a, a'))$, $N_G(B - r_B) \subseteq V(P)$ then
 8:     Find an up $a$-$a'$ $F$-chain $H$ by applying Lemma 3.6;
 9:     Return $H$ and stop;
10:   if $|X_P| = 2$ then
11:     Let $v, v'$ be the vertices in $X_P$;
12:     $H \leftarrow (G_F - (V(B_P) - X_P)) - vv'$;
13:     if $H$ is a down $F$-chain in $G$ then
14:       Return $H$ and stop;
15:     else
16:       Find a $B_P$-augmenting path $P'$ as in Lemma 3.7;
17:       Set $P \leftarrow P'$ and start a new iteration;
18:   if $|V(P)| = 3$ then
19:     Find a $B_P$-augmenting path $P'$ as in Lemma 3.8;
20:     Set $P \leftarrow P'$ and start a new iteration;
21:   Compute $\mathcal{H}$;
22:   Let $H_i \in \mathcal{H}$ be adjacent to $F$;
23:   if $x_i = y_i$ then
24:     Find a $B_P$-augmenting path $P'$ as in Lemma 3.12
25:     $P \leftarrow P'$ and start a new iteration;
26:   Let $H$ be obtained from $A_i$ by adding $N_G(A_i) \cap V(F)$ and all the edges of $G$
      from $V(A_i)$ to $V(F)$;
27:   if $G_F - V(B_P) = A_i$ and $H$ is a triangle chain of $F$ then
28:     Return $H$ and stop;
29:   else
30:     Find a $B_P$-augmenting path $P'$ as in Lemmas 3.12, 3.13, 3.14, and 3.15;
31:     Set $P \leftarrow P'$ and start a new iteration;
```

Now we discuss the running time of the algorithm.

The loop in line 1 is executed at most $|V(G)|$ times since $|V(B_P)|$ increases at each iteration.

By Lemma 3.4, the step in line 2 can be performed in $O(|V(G)| + |E(G)|)$ time.

The test in line 4 and the steps in line 6 can be executed in $O(|V(G)| + |E(G)|)$ time by standard graph search techniques [6].

The steps in lines 7–9 can be executed in $O(|V(G)| + |E(G)|)$ time by Lemma 3.6.

The steps in lines 10–17 can be executed in $O(|V(G)| + |E(G)|)$ time by Lemma 3.7.

The steps in lines 18–20 can be executed in $O(|V(G)| + |E(G)|)$ time by Lemma 3.8.

The steps in lines 21–22 can be executed in $O(|V(G)| + |E(G)|)$ time by standard graph search techniques [6].

The steps in lines 23–25 can be executed in $O(|V(G)|+|E(G)|)$ time by Lemma 3.12.

Finally, the steps in lines 26–31 can be executed in $O(|V(G)|+|E(G)|)$ time by Lemmas 3.12, 3.13, 3.14, and 3.15.

Therefore, the running time of the Algorithm 1 is $O(|V(G)||E(G)|)$. □

4. Chain decomposition.

In this section, we describe how to construct a nonseparating chain decomposition of a 4-connected graph G.

The idea is the following. Suppose we have found a partial chain decomposition $H_1, H_2, \ldots, H_{i-1}$ of G and we want to find the next chain H_i. Let $F := G[\bigcup_{j=1}^{i-1} I(H_j)]$, and assume that $G_F := G-(V(F)-\{r\})$ is 2-connected. If G_F is a planar cyclic chain rooted at r, then we obtain our desired decomposition by taking $H_i := G_F$ and $t := i$. If G_F is not a planar cyclic chain, then we want to use Theorem 3.2. In order to apply it, we need to efficiently find vertices $a, a' \in V(F)$ and a feasible a-a' F-path P. This will follow from Lemma 4.2 below.

We need the following result, proved in [7] and [1], which was used in [2].

THEOREM 4.1. *Let G be a 3-connected graph, let $e \in E(G)$, and let $u \in V(G)$ be nonincident to e. Then there exists a nonseparating induced cycle in G through e and avoiding u. Moreover, such a cycle can be found in $O(|V(G)|+|E(G)|)$ time.*

LEMMA 4.2. *Let G be a 4-connected graph, let $r \in V(G)$, and let F be a connected subgraph of G such that $r \in V(F)$, $|V(F)| \geq 2$, and $G_F := G-(V(F)-\{r\})$ is 2-connected. Then one of the following holds:*

(1) *G_F is a planar cyclic chain in G rooted at r; or*

(2) *there exists a feasible a-a' F-path P in G, that is,*

  (i) *$V(P) \cap V(F) = \{a, a'\}$ and P is an induced path in $G - aa'$;*

  (ii) *$P(a, a')$ is nonseparating in G_F;*

  (iii) *r is contained in a nontrivial block of $G_F - V(P(a, a'))$; and*

  (iv) *if $r \in \{a, a'\}$, then r is not a cut vertex of $G_F - V(P(a, a'))$.*

Moreover, one can in $O(|V(G)|+|E(G)|)$ time certify that (1) holds or find a path as in (2).

Proof. First, suppose that G_F is 3-connected. Let G' be obtained from G by contracting $F-r$ to a single vertex, say v'. Then G' is 4-connected; otherwise, there exists a 3-cut T in G'. Since G is 4-connected, $v' \in T$. But then $T-\{v'\}$ is a 2-cut in G_F, which is a contradiction. By Theorem 4.1, we can find a nonseparating induced cycle C in G' through rv' in $O(|V(G)|+|E(G)|)$ time. The path $C-rv'$ in G' corresponds to an induced path P in G from r to some vertex $a' \in V(F-r)$. Since G is 4-connected, r has at least two neighbors in $G_F - V(P(r, a'))$. Moreover, since C is nonseparating in G', r is not a cut vertex of $G_F - V(P(r, a'))$, and r is contained in a nontrivial block of $G_F - V(P(r, a'))$. Thus, P, $a := r$, and a' satisfy (2).

So we may assume that G_F is 2-connected but not 3-connected. Let $\{b, b'\}$ be a 2-cut of G_F. Let H_1, H_2 be edge-disjoint subgraphs of G_F such that $r \in V(H_1)$, $V(H_1) \cap V(H_2) = \{b, b'\}$, $H_1 \cup H_2 = G_F$, $|V(H_1)| \geq 3$, and $|V(H_2)| \geq 3$. Choose H_1, H_2 such that H_2 is minimal. Note that b, b', H_1, H_2 can be found in $O(|V(G)|+|E(G)|)$ time using the algorithm in [3] for finding the 3-connected components of G_F. Let $S := N_G(H_2 - \{b, b'\}) - \{b, b'\}$, and let G' be obtained from H_2 by adding S and the edges of G from S to $V(H_2)-\{b, b'\}$. Note that $S \subseteq V(F)$, $|S| \geq 2$, and $r \notin S$ because $\{b, b'\}$ is a 2-cut of G_F and $r \notin V(H_2) - \{b, b'\}$. Moreover, G' is $(4, S \cup \{b, b'\})$-connected.

Suppose that $|V(H_2)| \geq 4$. Then by minimality of H_2, H_2 is 2-connected and G', b, b', S satisfy (i)–(v) of Lemma 2.3 (with G' as G). Therefore, we can in $O(|V(G')|+|E(G')|)$ time either

(I) find $a, a' \in S$ and an induced a-a' path P' in G' such that $V(P') \cap \{b, b'\} = \emptyset$, $V(P') \cap S = \{a, a'\}$, and $G - (V(P') \cup S)$ is connected, or

(II) certify that $|S| = 2$, and the vertices in S can be labeled as a, a' such that (G', a, b, a', b') is planar.

If (I) occurs, then r is contained in a nontrivial block of $G - V(P')$ since there exists a b-b' path in $H_2 - V(P(a, a'))$. Since $r \notin S$, we have $r \notin \{a, a'\}$. Hence, $P := P'$ is a path that satisfies (2).

So we may assume that one of the following holds: $|V(H_2)| \geq 4$ and (II) occurs, or $|V(H_2)| = 3$.

We claim that one can find in $O(|V(G')| + |E(G')|)$ time a path P in G' with ends a, a' in S such that $G' - (V(P) \cup S)$ has exactly two components K, K' with $b \in V(K)$ and $b' \in V(K')$. If $|V(H_2)| \geq 4$ and (II) occurs, then this follows from Lemma 2.6. If $|V(H_2)| = 3$, then let v be the only vertex in $V(H_2) - V(H_1)$. Then v has degree two in G_F, and since G is 4-connected, v has at least two neighbors in $V(F)$, say a, a'. Then $P := (a, v, a')$ is the required path.

Therefore, $G_F - V(P(a, a'))$ is connected. If r is contained in a nontrivial block of H_1, then r is contained in a nontrivial block of $G_F - V(P(a, a'))$, and since $r \notin S$, $r \notin \{a, a'\}$. In this case, P satisfies (2).

So assume that r is contained only in trivial blocks of H_1.

Since G_F is 2-connected, H_1 is a b-b' chain. Moreover, either r is a cut vertex of H_1, or $r \in \{b, b'\}$. In both cases, G_F is a cyclic chain rooted at r. Let $G_F := v_0 B_1 v_1 \cdots v_{k-1} B_k v_k$ for some integer $k \geq 2$ (where $v_0 = v_k = r$). Note that either $H_2 = B_j$ for some $1 \leq j \leq k$ (when $|V(H_2)| \geq 4$), or $H_2 = B_j \cup B_{j+1}$ for some $1 \leq j \leq k - 1$ where B_j, B_{j+1} are trivial (when $|V(H_2)| = 3$).

If all the B_i's are trivial, then G_F is a planar cyclic chain and (2) holds. So assume that not all B_i's are trivial. For each 2-connected B_i, let $S_i := N_G(B_i - \{v_{i-1}, v_i\}) - \{v_{i-1}, v_i\}$, and let G_i be obtained from B_i by adding S_i and the edges of G from S_i to $V(B_i)$. Then $S_i \subseteq V(F - r)$, because $\{v_{i-1}, v_i\}$ is a 2-cut of G_F, and $r \notin V(B_i) - \{v_{i-1}, v_i\}$. Note that G_i, S_i, v_{i-1}, v_i (as G, S, b, b', respectively) satisfy (i)–(v) of Lemma 2.3 because $G_i - S_i = B_i$ is 2-connected and G_i is $(4, S_i \cup \{v_{i-1}, v_i\})$-connected. Thus, one can in $O(|V(G_i)| + |E(G_i)|)$ time either (a) find $a_i, a'_i \in S_i$ and an induced a_i-a'_i path P_i in G such that $V(P_i) \cap \{v_{i-1}, v_i\} = \emptyset$, $V(P_i) \cap S_i = \{a_i, a'_i\}$, and $G_i - (V(P_i) \cup S_i) = B_i - V(P_i(a_i, a'_i))$ is connected, or (b) certify that $|S_i| = 2$, and the vertices in S_i can be labeled as a_i, a'_i such that $(G_i, v_{i-1}, a_i, v_i, a'_i)$ is planar. Since G is 4-connected, if (b) occurs, then $B_i - \{v_{i-1}, v_i\} = G_i - (S_i \cup \{v_{i-1}, v_i\})$ is connected.

If G_F is not a planar cyclic chain rooted at r, then (a) must hold for some 2-connected B_i, and hence, $P := P_i$ satisfies (2) (because $r \notin S_i$). Otherwise, (1) holds.

It is not hard to see that all the steps described above can be executed in $O(|V(G)| + |E(G)|)$ time. □

Thus, combining Lemma 4.2 and Theorem 3.2 we obtain the following.

THEOREM 4.3. *Let G be a 4-connected graph, let F be a subgraph of G, and let $r \in V(F)$ such that $G_F := G - (V(F) - \{r\})$ is 2-connected. Then one of the following holds:*

(1) *there exists a good F-chain H in G such that $G_F - I(H)$ and $G[V(F) \cup I(H)]$ are 2-connected; or*

(2) *G_F is a planar cyclic chain rooted at r.*

Moreover, one can in $O(|V(G)|+|E(G)|)$ *time find a good* F*-chain as in* (1) *or certify that* (2) *holds.*

We are now ready to prove the main result in this paper.

Proof of Theorem 1.5. A nonseparating chain decomposition of G starting at ra can be found as follows. The first chain H_1 can be found in $O(|V(G)||E(G)|)$ time by Theorem 1.6. The internal chains can be found iteratively as follows. Suppose we have found a partial chain decomposition $H_1, \ldots, H_{i-1}$ ($i \geq 2$) of G and we want to find H_i. Let $F := G[\bigcup_{j=1}^{i-1} I(H_j)]$. Apply Theorem 4.3 to G, F, and r. Then one of the following holds:

(1) there exists a good F-chain H in G such that $G_F - I(H)$ and $G[V(F) \cup I(H)]$ are 2-connected; or

(2) G_F is a planar cyclic chain rooted at r.

Moreover, one can in $O(|V(G)|+|E(G)|)$ time find a planar chain as in (1) or certify that (2) holds. If (1) holds, then let $H_i := H$ and set $i \leftarrow i+1$. If (2) holds, then $H_1, \ldots, H_i := G_F$ is the desired chain decomposition.

Since the number of chains is at most $|V(G)|$, the above algorithm has time complexity $O(|V(G)|^2|E(G)|)$. □

Acknowledgment. The authors thank the referees for carefully reading the manuscript and for their helpful suggestions.

REFERENCES

[1] J. Cheriyan and S. N. Maheshwari, *Finding nonseparating induced cycles and independent spanning trees in* 3*-connected graphs*, J. Algorithms, 9 (1988), pp. 507–537.

[2] S. Curran, O. Lee, and X. Yu, *Nonseparating planar chains in* 4*-connected graphs*, SIAM J. Discrete Math., 19 (2005), pp. 399–419.

[3] J. E. Hopcroft and R. E. Tarjan, *Dividing a graph into triconnected components*, SIAM J. Comput., 2 (1973), pp. 135–158.

[4] J. E. Hopcroft and R. E. Tarjan, *Efficient planarity testing*, J. Assoc. Comput. Mach., 21 (1974), pp. 549–568.

[5] W. L. Hsu and W. K. Shih, *A new planarity test*, Theoret. Comput. Sci., 223 (1999), pp. 179–191.

[6] R. E. Tarjan, *Data Structures and Network Algorithms*, CBMS-NSF Regional Conf. Ser. in Appl. Math. 44, SIAM, Philadelphia, PA, 1983.

[7] W. T. Tutte, *How to draw a graph*, Proc. London Math Soc. (3), 13 (1963), pp. 743–767.

SIAM J. DISCRETE MATH.
Vol. 19, No. 4, pp. 881–899

POLYLOGARITHMIC ADDITIVE INAPPROXIMABILITY OF THE RADIO BROADCAST PROBLEM*

MICHAEL ELKIN† AND GUY KORTSARZ‡

Abstract. The input for the radio broadcast problem is an undirected n-vertex graph G and a source node s. The goal is to send a message from s to the rest of the vertices in the minimum number of rounds. In a round, a vertex receives the message only if exactly one of its neighbors transmits. The radio broadcast problem admits an $O(\log^2 n)$ approximation [I. Chlamtac and O. Weinstein, in Proceedings of the IEEE INFOCOM, 1987, pp. 874–881; D. Kowalski and A. Pelc, in APPROX-RANDOM, Lecture Notes in Comput. Sci. 3122, Springer, Berlin, 2004, pp. 171–182].

In this paper we consider the additive approximation ratio of the problem. We prove that there exists a constant c so that the problem cannot be approximated within an additive term of $c \log^2 n$, unless $NP \subseteq BTIME(n^{O(\log \log n)})$.

Key words. approximation, broadcast, radio

AMS subject classification. 68W25

DOI. 10.1137/S0895480104445319

1. Introduction.

1.1. The radio broadcast problem.

1.1.1. Definition and motivation. Consider a synchronous network of processors that communicate by transmitting messages to their neighbors, where a processor receives a message in a given step if and only if *precisely one* of its neighbors transmit. The instance of the radio broadcast problem, called *radio network*, is a pair $(G = (V, E), s)$, $s \in V$, where G is an unweighted undirected n-vertex graph and s is a vertex, called *source*. The objective is to deliver one single message that the source s generates to all the vertices of the graph G using the smallest possible number of communication rounds. The prescription that tells each vertex when it should broadcast is called the *schedule*; the *length* of the schedule is the number of rounds it uses, and it is called *feasible* if it informs all the vertices of the graph. From practical perspective, the interest in radio networks is usually motivated by their military significance as well as by the growing importance of cellular and wireless communication (see, e.g., [18, 14, 4]). The radio broadcast is perhaps the most important communication primitive in radio networks, and it has been intensively studied starting in the mid-1980s [9, 19, 20, 6, 5, 8, 17, 14, 18, 1, 4, 10, 7].

From theoretical perspective, the study of the radio broadcast problem provided researchers with a particularly convenient playground for the study of such broad and fundamental complexity-theoretic issues as the power and limitations of randomization and for the study of different models of distributed computation [4, 18, 20]. In this paper we study the approximation threshold of the *radio broadcast* problem.

*Received by the editors July 27, 2004; accepted for publication (in revised form) May 13, 2005; published electronically December 30, 2005.
http://www.siam.org/journals/sidma/19-4/44531.html

†Department of Computer Science, Ben-Gurion University, Beer-Sheva, Israel (elkinm@cs.bgu.ac.il). This work was done at the Department of Computer Science, Yale University, New Haven, CT 06510.

‡Computer Science Department, Rutgers University, Camden, NJ 08102 (guyk@crab.rutgers.edu). Part of this work was done at the School of Mathematics, Institute for Advanced Study, Princeton, NJ, 08540.

We believe that our results show that this problem is of particular interest from the standpoint of the theory of hardness of approximation as well.

1.1.2. Previous results.

Upper bounds. The first algorithm for the radio broadcast problem was devised by Chlamtac and Weinstein in 1987 [10]. That algorithm, given an instance (G, s) of the problem, constructs a feasible broadcast schedule of length $O(Rad(G, s) \cdot \log^2 n)$, where $Rad(G, s)$ stands for the *radius* of the instance (G, s), that is, the maximum distance $d_G(s, v)$ in the graph G between the source s and some vertex $v \in V$. Their algorithm is *centralized*; i.e., it accepts the entire graph as input.

Soon afterwards Bar-Yehuda, Goldreich, and Itai [4] devised a distributed randomized algorithm that provides feasible schedules of length $O(Rad(G, s) \cdot \log n + \log^2 n)$. Recently [21] a *deterministic* (albeit, centralized) algorithm with the same performance was given by Kowalski and Pelc. Alon et al. [1] have shown that the additive term of $\log^2 n$ in the result of [4, 21] is inevitable, and they devised a construction of infinitely many instances (G, s) of constant radius that satisfy that any broadcast schedule for them requires $\Omega(\log^2 n)$ rounds. Kushilevitz and Mansour [18] have shown that for *distributed* algorithms the multiplicative logarithmic term in the result of [4] is inevitable as well, and they proved that for *any distributed algorithm* for the radio broadcast problem there exists (infinitely many) instances (G, s) on which the algorithm constructs a schedule of length $\Omega(Rad(G, s) \cdot \log(n/Rad(G, s)))$. Finally, the gap between the $\log n$ and $\log(n/Rad(G, s))$ was recently closed by Kowalski and Pelc [20] and by Czumaj and Rytter [9].

Gaber and Mansour [14] devised a centralized algorithm that constructs feasible schedules of length $O(Rad(G, s) + \log^5 n)$. In [12] we improved this result providing a schedule of length $Rad(G, s) + O(\sqrt{Rad(G, s)} \cdot \log^2 n) = O(Rad(G, s) + \log^4 n)$.

Since, obviously, any schedule for an instance (G, s) requires at least $Rad(G, s)$ rounds, the algorithms for the radio broadcast problem [10, 4, 14, 20, 9, 21] can be interpreted as *approximation algorithms* for the problem. In particular, [10, 21] is a deterministic $O(\log^2 n)$ approximation algorithm.

Lower bounds. The NP-hardness of the radio broadcast problem was shown by Chlamtac and Kutten [7] as early as in 1985. In [15] an NP-hardness result is established for solving the problem on unit disk graphs.

A gap reduction is a reduction that maps an arbitrary NPC problem to the problem at hand giving some gaps for the optimum values resulting from a yes and a no instance. The authors of the current paper have shown [12] a gap reduction that maps a yes instance to a radio broadcast instance that admits a 3 rounds schedule, while a no instance is mapped into an $\Omega(\log n)$ schedule. This proves that there exists a constant $c > 0$ such that the radio broadcast problem cannot be approximated within approximation ratio of $c \log n$ unless $NP \subseteq BPTIME(n^{O(\log \log n)})$.

1.2. Our results. Note that [12, 14] can be considered as additive approximation algorithms for the problem. Hence, we study the additive ratio of radio broadcast. We provide a gap reduction that maps a yes instance to a radio network that admits a schedule of length $O(\log n)$ and maps a no instance to a radio network for which any feasible schedule is of length $\Omega(\log^2 n)$. Thus, there exists some $c > 0$ so that the radio broadcast problem admits no polynomial additive $c \log^2 n$ ratio approximation unless $NP \subseteq BPTIME(n^{O(\log \log n)})$. This fully determines the additive approximation ratio of the problem for graphs with radius at most $\log n$ as the result of [21] implies that, for graphs with radius at most $\log n$, there exists a matching additive upper bound of $O(\log^2 n)$. We are not aware of any other problem that exhibits a tight additive polylogarithmic ratio. (See [16, 13] for the only example of an almost tight polylogarithmic *multiplicative* approximation of which we are aware. This example is the group Steiner problem on trees.)

Remark. A big challenge seems to be designing a gap reduction that maps a yes instance to a constant number of rounds schedule and a no instance to a schedule of length $\Omega(\log^2 n)$. We leave this question open. If such a proof is possible, then the (multiplicative) best approximation ratio for the problem is $\log^2 n$ (up to constants) much like the group Steiner problem on trees. Alternatively, the challenge is to design an $O(\log n)$ ratio approximation for small radios graphs.

2. Preliminaries.

2.1. Definitions and notation.

We start by introducing some definitions and notations. In all the notations, we may eventually omit some parameters if the meaning can be deduced from the context.

Definition 2.1. *The set of neighbors of a vertex v in an unweighted undirected graph $G(V, E)$, denoted $\Gamma_G(v)$, is the set $\{u \in V \mid (v, u) \in E\}$. For a subset $X \subseteq V$, the set of neighbors of the vertex v in the subset X, denoted $\Gamma_G(v, X)$, is the set $\{u \in X \mid (v, u) \in E\}$.*

Notation 2.2. *For a positive integer number n, let $[n]$ denote the set $\{1, 2, \ldots, n\}$.*

Definition 2.3. *Let $G = (V, E)$ be an unweighted undirected graph, and let $R \subseteq V$ be a subset of vertices. The set of vertices* informed *by R, denoted $I(R)$, is $I(R) = \{v \mid \exists! x \in R \text{ such that } v \in \Gamma_G(x)\}$ (the notation $\exists! x$ stands for "there exists a unique x"). For a singleton set $R = \{x\}$, $I(R) = I(\{x\}) = I(x) = \Gamma_G(x)$.*

A sequence of vertex sets $\Pi = (R_1, R_2, \ldots, R_q)$, $q = 1, 2, \ldots$, is called a *radio broadcast schedule* (henceforth referred to as a *schedule*) if $R_{i+1} \subseteq \bigcup_{j=1}^{i} I(R_j)$ for every $i = 1, 2, \ldots, q - 1$. Intuitively, this condition means that the vertices that send a message in a certain round have to be informed in one of the previous rounds.

The set of vertices *informed by a schedule* Π, denoted $I(\Pi)$, is $I(\Pi) = \bigcup_{R \in \Pi} I(R)$.

Given a graph $G = (V, E)$ and a vertex $s \in V$, a schedule Π is *feasible* with respect to (G, s) if $R_1 = \{s\}$ and $V \subseteq I(\Pi)$.

The *length* of the schedule $\Pi = (R_1, R_2, \ldots, R_q)$ is $|\Pi| = q$.

An instance of the *radio broadcast problem* $\mathcal{G}$ is a pair $(\bar{G} = (\bar{V}, \bar{E}), s)$, where $\bar{G}$ is a graph and $s \in \bar{V}$ is a vertex. The goal is to compute a feasible schedule Π of minimal length. The *value* of an instance $\mathcal{G}$ of the radio broadcast problem is the length of the shortest feasible schedule Π for this instance.

For any schedule $\Pi = (R_1, R_2, \ldots, R_q)$, the set R_i is called the *ith round* of Π, $i = 1, 2, \ldots, q$.

2.2. The MIN-REP problem.

Definition 2.4. *The MIN-REP problem is defined as follows. The input consists of a bipartite graph $G = (V_1, V_2, E)$. In addition, for $j = 1, 2$, the input contains a partition $\tilde{V}_j$ of V_j into a disjoint union of subsets, $V_1 = \bigcup_{A \in \tilde{V}_1} A$, $V_2 = \bigcup_{B \in \tilde{V}_2} B$. The triple $\mathcal{M} = (G, \tilde{V}_1, \tilde{V}_2)$ is an* instance *of the MIN-REP problem. The size of the instance is $n = |V_1| + |V_2|$. An instance G as above induces a bipartite supergraph $\tilde{G} = (\tilde{V}_1, \tilde{V}_2, \tilde{E})$ in which the sets A and B of the partition serve as the vertices of the supergraph. The edges of the supergraph are $\tilde{E}(\mathcal{M}) = \tilde{E} = \{(A, B) \in \tilde{V}_1 \times \tilde{V}_2 \mid a \in A, b \in B, (a, b) \in E\}$. In other words, there is a (super)edge between a pair of sets $A \in \tilde{V}_1$, $B \in \tilde{V}_2$ if and only if the graph G contains an edge between a pair of vertices $a \in A$, $b \in B$.*

Denote $\tilde{V} = \tilde{V}_1 \cup \tilde{V}_2$. A pair of vertices $x_1, x_2 \in V_1 \cup V_2$ is called a matched pair with respect to a superedge $\tilde{e} = (A, B) \in \tilde{E}$ *(henceforth,* $\tilde{e}$-m.p.*) if $(x_1, x_2) \in E$ and either $x_1 \in A$ and $x_2 \in B$ or vice versa.*

A subset $C \subseteq V_1 \cup V_2$ of vertices is said to cover *a superedge $\tilde{e} = (A, B)$ if it contains an $\tilde{e}$-m.p. A subset $C \subseteq V_1 \cup V_2$ that satisfies $|C \cap X| = 1$ for every $X \in \tilde{V}$ is called a* MAX-cover*. In other words, a MAX-cover C contains exactly one vertex from each supervertex.*

An instance $\mathcal{M}$ of the MIN-REP problem is called a yes instance *if there exists a MAX-cover that covers all the superedges. Such a MAX-cover is called a* perfect MAX-cover*.*

*For a positive real number $t > 1$, an instance $\mathcal{M}$ of the MIN-REP problem is called a t-*no instance *if any C that covers at least half of the superedges must pick on average at least t elements of every A and every B (every C that covers at least half of the superedges has size at least t times the number of A, B sets).*

The maximization version of MIN-REP problem is equivalent to the maximization variant of the label-cover problem (see, e.g., [3]).

The parameters of the MIN-REP instance. We impose several additional (somewhat less standard) restrictions on the set of instances of the MIN-REP problem. For the rest of the paper, let n denote the number of vertices in the MIN-REP instance.

1. All the supervertices $X \in \tilde{V}$ are of size polylogarithmic in n (namely, the size at most $(\log n)^d$ for some constant d).
2. The number of superedges is $O(n \cdot \ polylog(n))$.
3. *The star property.* For every superedge $\tilde{e} = (A, B) \in \tilde{E}$, $A \subseteq V_1$ and $B \subseteq V_2$, and every vertex $b \in B$, there exists exactly one vertex $a \in A$, denoted $\tilde{e}(b)$, such that $(a, b) \in E$. The set of all vertices b such that $\tilde{e}(b) = a$ for the same vertex a, along with the vertex a, is called an *$\tilde{e}$-star*.

Essentially, the star property means that for every superedge $\tilde{e} = (A, B) \in \tilde{E}$ the graph induced by the subset $A \cup B$ decomposes into a collection of vertex-disjoint stars (a *star* is a graph with all vertices but (maybe) one having degree 1). The vertex with degree larger than 1 is called the *head* of the star; if there are only two vertices in the star, then the head is the vertex that belongs to the supervertex A. The other vertices of the star are called the *leaves* of the star. See Figure 1 for an example of a MIN-REP instance that obeys the star property.

THEOREM 2.5 (see [2, 23]). *No deterministic polynomial time algorithm can distinguish between the yes-instances and the $\log^{10} n$-no instances of the MIN-REP problem unless $NP \subseteq DTIME(n^{O(\log \log n)})$, even when the instances of the MIN-REP problem satisfy the conditions* (1)–(3)*.*

3. The construction.

3.1. The high-level idea.

In [22] a reduction from an arbitrary NPC language to the set-cover problem is given. The elements of the set-cover instance are grouped into a union of *ground sets*. In a yes instance, every ground set M can be covered by two "complementary" sets each covering a disjoint half of M. In a no instance, every set in the set-cover instance that contains elements of M (essentially) contains a random half of M, and so $\Omega(\log |M|)$ sets are required to cover the entire set M. This is used in [11] to design a radio network that admits no schedule of length $o(\log n)$ for a no instance, and a schedule with only a constant number of rounds for a yes instance.

In [1], another special kind of set-cover is designed. The elements are partitioned to $\Theta(\log n)$ ground sets M_j. In this instance, covering *uniquely* many elements in $\bigcup M_j$

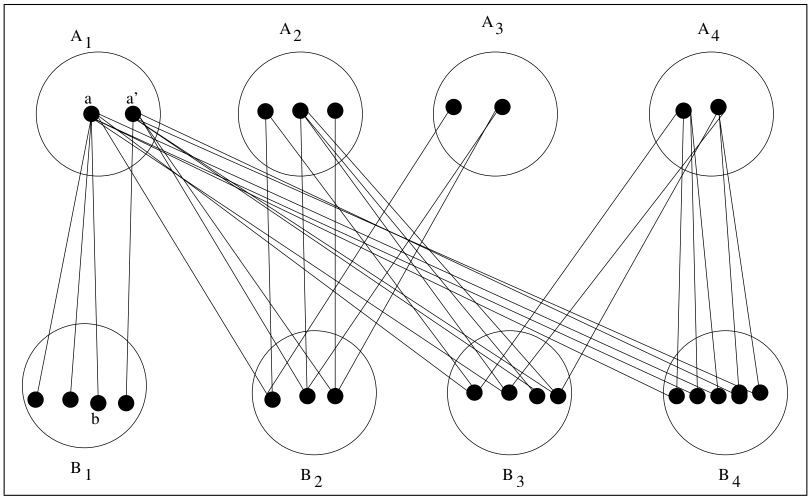


FIG. 1. *An example of a MIN-REP instance that satisfies the star property. Every pair of sets (A_i, B_j) in the partition induces a collection of disjoint stars with heads in A. The vertices (a, b) form a matching pair that covers the superedge (A_1, B_1). The vertices a', b do not form a matching pair and do not cover the superedge (A_1, B_1).*

is not possible (an element is uniquely covered by a collection of sets if it belongs to exactly one set in the collection). Specifically, if a collection of sets uniquely covers "many" M_j elements, then it does not uniquely cover many elements of M_q for any $q \neq j$. This construction is used in [1] to design a radio network that, essentially, has to inform the sets M_j "one by one" while the construction for every M_j is similar to the one in [22, 11], namely, informing M_j by itself requires $\Omega(\log n)$ rounds. Since the number of sets M_j is $\Theta(\log n)$, a lower bound of $\Omega(\log^2 n)$ for the length of a feasible schedule follows.

We modify the construction of [1] and add a"trapdoor" to their construction, using ideas of [22]. This trapdoor makes it possible to inform every M_j in 2 rounds. This guarantees that for a yes instance, a feasible schedule of logarithmic length exists: simply inform M_j one j after the other. On the other hand, the modification maintains the lower bound of $\Omega(\log^2 n)$ for a no instance. Hence, we obtain a gap between $O(\log n)$ and $\Omega(\log^2 n)$, that is, an additive gap of $\Omega(\log^2 n)$.

3.2. The construction of [1].

Since our construction relies on that of [1], we briefly sketch their construction.

DEFINITION 3.1. *A schedule of at most* $\log^2 n/100$ *rounds is called a short schedule.*

Let $(\mathcal{X}, \mathcal{Y})$ be two sets of vertices. Let $|\mathcal{X}| = n$ and $\mathcal{Y}$ be a disjoint union $\mathcal{Y} = \bigcup \mathcal{Y}_j$ of sets $\mathcal{Y}_j$ each containing n^7 vertices for $0.4 \cdot \log_2 n \leq j \leq 0.6 \cdot \log_2 n$. Thus, $|\mathcal{Y}| = \Theta(n^7 \log n)$.

A vertex $x \in \mathcal{X}$ and a vertex $y \in \mathcal{Y}_j$ are connected with an edge with probability $1/2^j$ independently of other edges. In addition, add a source s and connect s to all the vertices of $\mathcal{X}$. Observe that, by definition, after the first round the set of informed vertices is exactly $\{s\} \cup \mathcal{X}$.

Intuitively, in the above construction any transmitting subset of $S \subseteq \mathcal{X}$ helps to inform only part of the sets $\mathcal{V}_j$. It is not possible to choose a size for S so that all $\mathcal{Y}_j$ will contain many vertices informed by S. For a given j, for any set S of size larger than 2^j, there may exist many vertices of $\mathcal{Y}_j$ having at least two neighbors in S. But if S is much smaller than 2^j, then many of the vertices of $\mathcal{Y}_j$ will not have even a single neighbor in S.

The following elegant lemma formalizes this intuition.

LEMMA 3.2 (see [1]). *Let* $\Pi = (R_1, R_2, \ldots, R_t)$ *be a short (namely,* $t \le \log^2 n/100$*) collection of subsets of* $\mathcal{X}$*. Then there exists a subset* $S \subseteq \mathcal{X}$ *and an index* j*,* $0.4 \cdot \log_2 n \le j \le 0.6 \cdot \log_2 n$*, so that the following hold.*

1. $|S| \le 2^j \cdot \log_2 n$.
2. *Let* $\Phi' = (R'_1, R'_2, \ldots) = \Phi \setminus S$*. Then for each round* R'_q *in the schedule,* $|R'_q| \ge 2^j$*.*
3. *Let* f_k *be the number of sets in the schedule with cardinality* $2^{j+k} \le |R'_j| \le 2^{j+k+1}$*. Then,*

$$\sum_{k \ge 0} \frac{f_k}{2^k} \le \log n.$$

Indeed, how could a "short" schedule Φ cover $\mathcal{Y}_j$? The set S has size 2^j, so there is a nonnegligible probability that no vertex in S is connected to a vertex in $\mathcal{Y}_j$. (We use the term *nonnegligible* for a probability which is at least $\frac{1}{poly(n)}$, where $poly(n)$ is some polynomial in n.) Thus, the task of covering $\mathcal{Y}_j$ may be left to $\Phi' = (R'_1, R'_2, \ldots) = \Phi \setminus S$. Consider some vertex $y_j \in \mathcal{Y}_j$. Observe that, by item 2 above, each R'_q is of large enough size to make the probability of R'_q not informing y_j nonnegligible. Indeed, it is reasonable to expect that at least two vertices of $\mathcal{R}'_q$ will be connected to y_j in which case y_j does not get the message in round q. Now, since $\mathcal{Y}_j$ is "large" (has size n^7), there is a high probability that there will be a vertex that is not going to be informed at any round. The paper of Alon et al. [1] formally proves this claim along these lines.

3.3. Intuition behind the random permutation step. One of the difficulties in imitating the construction of [22] and combining it with the construction of [1] is as follows. The construction of [1] requires that vertices are connected to $\mathcal{Y}_j$ with probability $1/2^j$. On the other hand, in the construction of [22] some vertices are connected to one half of the elements in every ground set M_j (see [22] for more details). Thus, the probability that a and $v \in M_j$ are connected is $1/2$.

The way we overcome this difficulty is by forming *many* copies of every vertex $x \in A \cup B$.

Consider some superedge $\tilde{e}$ and the ground sets that correspond to $\tilde{e}$. Suppose that $M = M_{\tilde{e}}(j)$ is some ground set that corresponds to $\tilde{e}, j$ (similar to $\mathcal{Y}_j$ but dedicated to $\tilde{e}$). Every copy of a is connected in M to some random subset of size $|M|/2^{j+1}$. We ensure that neighbors in M of different a-copies are disjoint and that 2^j copies of a take part in this process. This implies that altogether the copies of $a \in A$ are indeed connected to a half of M. Let M_a denote this half.

In addition, let (a, b) be an $\tilde{e}$-matching pair. Then the copies of b are similarly connected but to $M_j \setminus M_a$. Thus, the copies of b cover the complementary half of M.

This way we are able on the one hand to control the degrees of copies of a and b (that is, to make it roughly $|M|/2^j$, as required in [1]) but on the other hand to guarantee that the copies of a and b cover together disjoint halves of M (as required in [22]).

For the claims in [1] to work we need the neighbors of (copies of) a and b in M to be random. We use copies of a and b for covering random elements of M as follows.

We first choose a random half of M. Then this random half is arbitrarily split into 2^j equal parts. Then we match the 2^j copies of a and the 2^j parts by a random permutation. The copy of a is connected to all vertices in its matching part. Similar construction is applied for copies of b on the complementary half.

3.4. The random permutation step: Formal definition. For the rest of the paper, let n denote the number of vertices of the MIN-REP graph. Consider an instance $\mathcal{M} = (G, \tilde{V}_1, \tilde{V}_2)$, $G = (V_1, V_2, E)$ of the MIN-REP problem with $V_1 = \bigcup_{A \in \tilde{V}_1} A$, $V_2 = \bigcup_{B \in \tilde{V}_2} B$. The reduction constructs an instance $\mathcal{G} = \mathcal{G}(\mathcal{M}) = (\bar{G}, s)$, $\bar{G} = (\bar{V}, \bar{E})$, $s \in \bar{V}$, of the radio broadcast problem in the following way.

Let $N = n^{0.6}$. The vertex set $\bar{V}$ of the graph consists of the source s, and the disjoint vertex sets $\mathcal{V}_1$ and $\mathcal{V}_2$.

The vertex set $\mathcal{V}_1$ contains N copies of every vertex a or b in $V = V_1 \cup V_2$; the set of all copies of a vertex x ($x = a$ or $x = b$) is denoted by $cp(x)$, and $cp(x, j)$ is the subset that contains the first 2^j copies of x.

For a subset $X \subseteq V$, let $cp(X)$ denote $cp(X) = \bigcup_{x \in X} cp(x)$. Let $\hat{J}$ denote the set of indices $\{0.4 \log n, 0.4 \log n + 1, \ldots, 0.6 \log n\}$.

The vertex set $\mathcal{V}_2$ is of the form $\mathcal{V}_2 = \bigcup_{\tilde{e} \in \tilde{E}} M_{\tilde{e}}$, where the *ground sets* $M_{\tilde{e}}$ are disjoint and all have equal size. Each ground set $M_{\tilde{e}}$ is a disjoint union of the sets $M_{\tilde{e}}(j, q)$, $j \in \hat{J}$, $q \in [L]$, with $L = n^{c_0+4}$, and c_0 is an integer positive universal constant that will be determined later. The sets $M_{\tilde{e}}(j, q)$ are all of equal size $M = n^{c_0}$ for the same constant c_0.

The edge set $\bar{E}$ of the graph $\bar{G}$ contains edges that connect the source s to the vertices of the set $\mathcal{V}_1$. We next construct the edge set between the vertices of $\mathcal{V}_1$ and $\mathcal{V}_2$. Fix $\tilde{e} = (A, B) \in \tilde{E}$ and the indices $j \in \hat{J}$, $q \in [L]$.

The random permutation step.

1. For every star head $a \in A$ let $M_a = M_{\tilde{e},a}(j, q)$ be an *exact random half* of the set $M = M_{\tilde{e}}(j, q)$.
2. For every vertex b in the star of a set $M_b = M_{\tilde{e},b}(j, q) = M \backslash M_a$.
   Remark. Steps 1 and 2 will be referred to as the *exact partition step.*
3. For every vertex $a \in A$, partition the set $M_a = M_{\tilde{e},a}(j, q)$ arbitrarily into 2^j disjoint subsets of equal size. Randomly permute $cp(a, j)$ (the first 2^j copies of a), and connect the ith copy (in the order determined by the random permutation) of a to the ith part of M_a.
4. Similarly, for every vertex b that belongs to the star of a, cover M_b by a random permutation. The random permutations of a and of leaves in the star of a are independent.

See Figure 2 for an illustration of the random permutation step.

Remarks.

1. The parameter q in $M = M_{\tilde{e}}(j, q)$ does not affect the probability of $\mathcal{V}_1$ vertices to be adjacent to $M_{\tilde{e}}(j, q)$. This probability is $1/2^j$. Unlike [1], many "$\mathcal{Y}_j$ type" sets are required. It is important though that if $q \neq q'$, then different events for $M_{\tilde{e}}(j, q)$ and $M_{\tilde{e}}(j, q')$ are independent.
2. If b and b' belong to the star of a in $\tilde{e}$, then $M_b = M_{b'} = M \setminus M_a$. However, the random permutations of b and b' are independent and are likely to be different.

Adding dummy vertices. Recall that $cp(X)$ is the set of all copies of X vertices. Currently, $|cp(A)|, |cp(B)| = \tilde{O}(n^{0.6})$. We need to later use Lemma 3.2 with $cp(A) \cup cp(B)$ playing the role of $\mathcal{X}$. For that, we need that $|cp(A) \cup cp(B)| = n$ (otherwise, it is required to use the lemma with $\tilde{\Theta}(n^{0.6})$ playing the role of n which may be

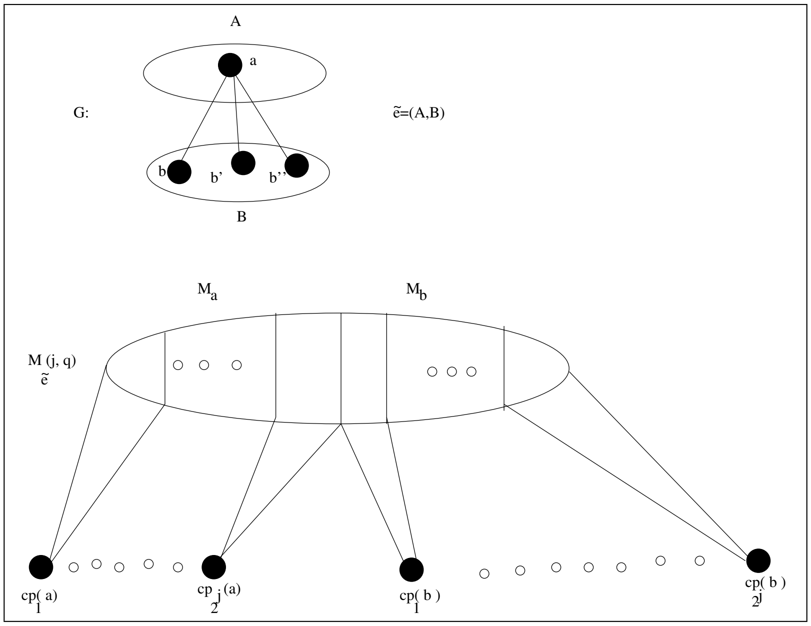


FIG. 2. *The figure illustrates the random permutation step. The copies of the vertex a cover an exact half of the vertices of the set $M_{\tilde{e}}$. The copies of the vertex b cover the complementary half of the vertices.*

confusing). Add dummy vertices to every $cp(A)$ and $cp(B)$ to complete its size to $n/2$ each. The dummy vertices have no connection to $\mathcal{V}_2$ but are joined to s. Thus, dummy vertices never transmit (do not belong to any round). This change affects only the constants in the ratio. Thus, we assume throughout that $|cp(A)| = |cp(B)| = n/2$ for every A, B.

3.5. The mixing step: Intuition and formal definition. Intuitively, we need to build a reduction in which a short schedule for the resulting radio network necessarily reveals a good solution for the original instance of the MIN-REP problem. Namely, a round is forced to use copies of many matched pairs; otherwise, the connections are random in a way similar to [1].

By the construction so far, this goal is not yet achieved because vertices can "coordinate efforts" even if they do not belong to a matching pair. For example, observe that if b, b' both belong to the star of a, then the copies of b and b' are connected in $M_{\tilde{e}}(j,q)$ to the same half (see Figure 2). This half is $M_b = M_{b'} = M_{\tilde{e}}(j,q) \setminus M_{\tilde{e},a}(j,q)$. Even though the random permutations of b and b' are independent, inserting both copies of b and b' into R increases the probability that no vertex in $M_b = M_{b'}$ remains uncovered. Therefore, so far we have not prevented b and b' from coordinating efforts.

Thus, we should modify the construction so that copies of b and copies of b' "hurt each other," and consequently, they cannot be used in the same round together to cover many vertices. This is done by adding some random edges. Copies of a are randomly connected to the other half of the vertices, namely, to the vertices of the set $M_b = M_{b'} = M_{\tilde{e}}(j,q) \setminus M_{\tilde{e},a}(j,q)$. Copies of b are randomly connected to $M_a = M_{\tilde{e},a}(j,q)$. See Figure 3.

These additional edges prevent a schedule from forming very large rounds. Because if a round is very large, many elements are covered two times or more. For example, inserting many copies of b and also many copies of b' into a round leads to "over-covering" vertices.

Further, the copies of $cp(a) \setminus cp(a,j)$ pose a problem. So far, they have no edges to $M_{\tilde{e}}(j,q)$. This would imply that we may add vertices from $cp(a) \setminus cp(a,j)$ without affecting $M_{\tilde{e}}(j,q)$, and consequently, it leaves a possibility of forming large big rounds with only a small number of vertices that are connected to $M_{\tilde{e}}(j,q)$.

Hence, we need to connect every $cp(a) \setminus cp(a,j)$ vertex to every $M_{\tilde{e}}(j,q)$ vertex with probability $1/2^j$.

3.6. The mixing step: Formal definition. Let $j, q, \tilde{e}$, $\tilde{e} = (A,B)$ be fixed. Fix some $\tilde{e}$-star with head a. Let $M = M_{\tilde{e}}(j,q)$. Let M_a be the set of neighbors of (the copies of) a in M and $M_b = M \setminus M_a$.

1. For every copy of a in $cp(a,j)$ and every vertex $v \in M_{\tilde{e},b}(j,q)$, add an edge between those two vertices with probability $1/2^j$.
2. Similarly, for every copy of b in $cp(b,j)$ and every vertex $v \in M_a = M_{\tilde{e},a}(j,q)$, add an edge with probability $1/2^j$.
3. For every j and every $x \in A \cup B$, connect every vertex of $cp(x) \setminus cp(x,j)$ to every vertex of $M_{\tilde{e},a}(j,q)$ independently, with probability $1/2^j$.

Steps 1, 2, and 3 will be referred to as the *mixing step.* See Figure 3.

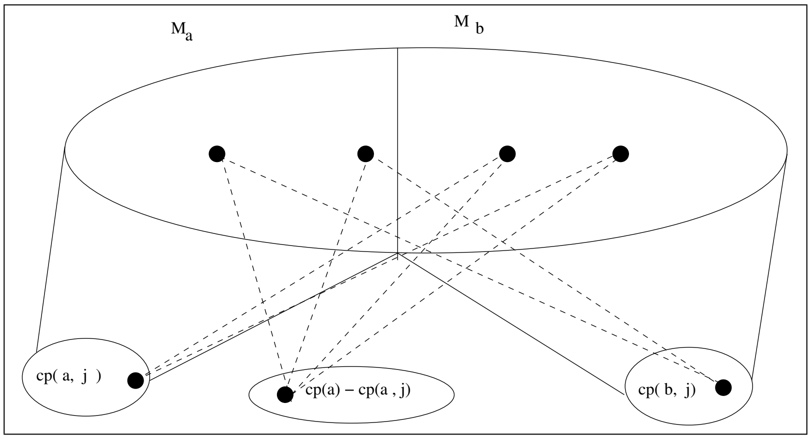


FIG. 3. *The figure illustrates the mixing step for some fixed q, j. The dotted edges represent random events that may result in edges. The probability for such an edge to be present is $1/2^j$. The figure also indicates that the vertices of $cp(a,j)$ form an exact cover of $M_a = M_{a,\tilde{e}}(j,q)$ by the random permutation step. On the other hand, pairs of vertices from $cp(a) \setminus cp(a,j)$ and M are connected with probability $1/2^j$ for every pair.*

3.7. Trapdoor: A schedule of logarithmic length for a yes instance. One way to explain some of the ideas behind the construction is by showing that the radio network resulting from a yes instance of the MIN-REP problem admits a schedule of length $O(\log n)$.

Let $\mathcal{M} = (G = (V,E), \tilde{G})$, $|V| = n$, be a yes instance of the MIN-REP problem, and let $(\mathcal{G}, s)$ be the instance of the radio broadcast problem that is obtained via our reduction.

Let C be a perfect MAX-cover, that is, a subset of the set $V_1 \cup V_2$ that covers all the superedges and contains exactly one vertex from each supervertex. (Recall that, by definition of the yes instance of the MIN-REP problem, there exists a perfect MAX-cover C for such an instance.)

Lemma 3.3. *There is a schedule of length $O(\log n)$ for the radio network $(\mathcal{G}, s)$.*

Proof. On the first round s transmits, and all the vertices of $\mathcal{V}_1$ are informed. Then, for each index j, $0.4 \log n \leq j \leq 0.6 \log n$, build two rounds. On the first one all the vertices of $\bigcup_{a \in C} cp(a, j)$ transmit in parallel, and on the second one all the vertices of $\bigcup_{b \in C} cp(b, j)$ transmit in parallel. Altogether, we obtain a schedule with $2 \cdot (0.2 \log n + 1) = O(\log n)$ rounds. □

Claim 3.1. *The schedule informs $\mathcal{V}_2$.*

Proof. Since the set C is a perfect MAX-cover, for every superedge $\tilde{e} = (A, B) \in \tilde{E}$, there exists some $\tilde{e}$-m.p. $a, b \in C$. Thus, when $cp(a, j)$ broadcasts, all the sets $M_{a,\tilde{e}}(j, q)$ are informed (observe that the mixing step does not insert edges between $cp(a, j)$ and $M_{\tilde{e},a}(j, q)$). Also, when $cp(b, j)$ transmits, all of the sets $M_{b,\tilde{e}}(j, q)$ are informed. □

4. Analysis, part I: Comparison to Lemma 3.2.

For the rest of the paper, let $a, b, b', \tilde{e}, j, q, M_{\tilde{e}}(j, q)$, and $v \in M_{\tilde{e}}(j, q)$ be vertices, indices, and sets that satisfy that (a, b) and (a, b') are $\tilde{e}$-matching pairs.

Since $j, q, \tilde{e}$ are fixed, for the simplicity of the notation we use M for $M_{\tilde{e}}(j, q)$ and M_a for $M_{\tilde{e},a}(j, q)$, etc. See also Figure 2.

In the next subsection we discuss a set $\mathcal{S}$ of size at most $2^j \cdot \ln n$. This set is analogous to the set S from Lemma 3.2. We need to estimate the probability that no element of $\mathcal{S}$ covers v (which we call the *probability of silence*). Later, we consider a subset $\mathcal{R}$ of size at least 2^j and discuss the probability that v has at least two neighbors in $\mathcal{R}$.

4.1. Probability of silence.

Lemma 3.2 shows that there exists a relatively small set S with useful properties. The probability that no element of S is connected to v is at least $1/n^{1+o(1)}$. In the construction of Alon et al. [1] computing this probability of silence is not difficult because each vertex of $\mathcal{Y}_j$ (for the same index j) is connected to v with probability $\frac{1}{2^j}$ *independently* of other vertices.

In our reduction it is *not necessarily true* that every small enough set $\mathcal{S}$ does not cover v with a nonzero probability. For example, suppose that $v \in M_a$ and $\mathcal{S}$ contains all the copies of a. Then, by definition of the random permutation step, the copies of a cover v *with probability 1.* Further, if $M_b = M \setminus \mathcal{S}$ and $\mathcal{S}$ contains all the copies of b, then $\mathcal{S}$ covers the entire set M with probability 1.

On the other hand, the cover of M that we have just described uses a matching pair (a, b). Thus, intuitively, our goal is to prove that any set $\mathcal{S}$ that does not use matching pairs does not cover v with a nonnegligible probability.

For the probability of the event "$\mathcal{S}$ does not cover v in the random permutation step" to be greater than zero, we need v to satisfy the following property.

The safety property. If $v \in M_x$, then $\mathcal{S}$ contains "only a fraction of" the copies of x. Alternatively, if all the copies of some x belong to $\mathcal{S}$, then $v \notin M_x$ must hold.

This is further formalized in the following definition.

Definition 4.1. *The partitions defined by the exact partition steps (namely, in steps* 1 *and* 2 *of the random permutation step) are* safe *for $\mathcal{S}$ and v if for every x, so that $v \in M_x$,*

$$|\mathcal{S} \cap cp(x, j)| \leq 2^j/8.$$

We shall see that if the partition is safe for $\mathcal{S}$ and v, then with a nonnegligible probability all the various random permutation steps do not cover v. The following lemma formalizes this claim.

We use the notation $\mathcal{S}\ \mathcal{AN}\ v$ to denote the event "no vertex of $\mathcal{S}$ is connected to v."

LEMMA 4.2. *Suppose that the partitions formed by the exact partition steps are safe for $\mathcal{S}$ and v. Suppose also $|\mathcal{S}| \leq 2^j \cdot \ln n$. Then*

$$\mathbb{P}\left((\mathcal{S}\ \mathcal{AN}\ v)\right) \geq \frac{1}{n^4}.$$

Proof. We use the following notation in the proof: S_N is the set of copies $x_q \in \mathcal{S}$ of some vertex x that *cannot* be connected to v in the random permutation step. Namely, either $v \in M \setminus M_x$ or $q > 2^j$ (x_q is not one of the 2^j first copies of x). The complement set is denoted by $S_Y = \mathcal{S} \setminus S_N$.

For every vertex $x \in \mathcal{S}$, let $S(x)$ be defined by $S(x) = cp(x,j) \cap S_Y$ (these are copies that can be connected to v by the random permutation step). Let $s(x) = |S(x)|$. Clearly, $s(x) \leq 2^j/8$. See an illustration for this notation in Figure 4.

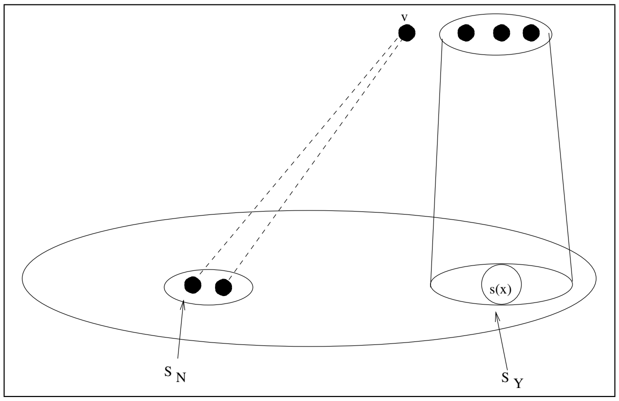


FIG. 4. *The effect of partition on M_x and $M \setminus M_x$, in addition to the random permutation step and mixing step on $\mathcal{S}, v$. The set $\mathcal{S}$ is partitioned to S_N and S_Y. The vertices of S_N cannot be connected to v by the random permutation step because v does not belong to "their half" of M. Only vertices that belong to the set S_Y may be connected to the vertex v by the permutation step. For x, vertices of $S(x)$ can be connected to v by the random permutation step. The figure illustrates the "silent scenario"; hence, the vertices of $S(x)$ are not connected to the vertex v by the random permutation step and all the dotted lines represent nonedges.*

The probability that no copy of x *that belongs to* $S(x)$ is connected to v in the random permutation step is

$$\frac{2^j - s(x)}{2^j}.$$

This is because 2^j copies of x participate in the random permutation step and only $s(x)$ of them belong to S_Y.

For two different vertices $x, y \in S_Y$, note that the choices of the random permutations of x and y are independent (this is true even if they are leaves in the same star). Let $\mathcal{A}$ be the event that S_Y does not cover v in the random permutation step. Hence,

$$\mathbb{P}(\mathcal{A}) \ \geq\ \Pi_{x\in\mathcal{S}}\frac{2^j - s(x)}{2^j} \geq \Pi_{x\in\mathcal{S}}\left(\left(1-\frac{s(x)}{2^j}\right)\cdot e^{-1}\right)^{s(x)/2^j}.$$

The last inequality is because for any positive real $u > 0$, $(1 - 1/u)^{u-1} \geq 1/e$, and so $1 - 1/u \geq ((1 - 1/u) \cdot e^{-1})^{1/u}$.

As $s(x) \leq 2^{j-3}$,

$$\left(1 - \frac{s(x)}{2^j}\right) \cdot e^{-1} \geq \frac{7}{8 \cdot e}.$$

Hence,

$$\Pr(\mathcal{A}) \geq \left(\frac{7}{8 \cdot e}\right)^{\sum_x s(x)/2^j} \geq \left(\frac{7}{8 \cdot e}\right)^{\ln n} \geq 1/n^2.$$

(The second inequality follows as $\sum_{x \in \mathcal{S}} s(x) \leq 2^j \cdot \log n$.)

The vertices of S_N are independently connected to v with probability $1/2^j$. (This follows from the mixing step of the reduction). Let $\mathcal{B}$ be the event that the mixing step does not form an edge between s and v. Hence, $\mathbb{P}(\mathcal{B}) \geq (1 - 1/2^j)^{|\mathcal{S}|} \geq (1 - 1/2^j)^{2^j \log n} \geq 1/n^2$. Finally, the event "no vertex of the set $\mathcal{S}$ is connected to the vertex v by the mixing step" is independent of the event "no vertex of the set $\mathcal{S}$ is connected to the vertex v by the permutation step." Hence,

$$\mathbb{P}(\mathcal{S}\ \mathcal{AN}\ v) \geq \frac{1}{n^2} \cdot \frac{1}{n^2} = \frac{1}{n^4}\ . \qquad \square$$

The probability for a safe partition. The partitions of M_x can be unsafe with probability 1 for some "problematic" sets $\mathcal{S}$. In fact, one can easily guarantee that v *is covered* by the random permutation step. This can be achieved by taking into $\mathcal{S}$ all copies of a and all copies of b (recall that (a, b) is an $\tilde{e}$-matching pair).

The following definition utilizes this idea.

Definition 4.3. *A set $\mathcal{S}$ is $(\tilde{e}, j)$-*partial *if for every $\tilde{e}$-m.p. (x, y), the set $\mathcal{S}$ contains at most 2^{j-3} vertices of $cp(x, j)$* or *it contains at most 2^{j-3} copies of $cp(y, j)$.*

If $\mathcal{S}$ is $(\tilde{e}, j)$-partial, it is still possible that after the coins are tossed in the exact partition step, v will not be covered by $\mathcal{S}$ in the random permutation step. Namely, we shall see that if $\mathcal{S}$ is $(\tilde{e}, j)$-partial, the partition is safe with a nonnegligible probability.

Definition 4.4. *Let $\tilde{e} = (A, B)$ be a superedge. Let $a \in A$. The set $star(a, \tilde{e})$ is the set of all copies of a and all copies of leaves b in an $\tilde{e}$-star of A.*

Lemma 4.5. *Let $\mathcal{S}$ be an $(\tilde{e}, j)$-partial of size $|\mathcal{S}| \leq 2^j \ln n$. The probability that the partition is safe for $\mathcal{S}, v$ is at least $1/n^8$.*

Proof. Since $|\mathcal{S}| \leq 2^j \cdot \ln n$ and the stars $star(a')$ with different vertices $a' \in A$ are all disjoint, the number of such stars that satisfy $|star(a) \cap \mathcal{S}| \geq 2^{j-3}$ is at most $8 \cdot \ln n$. Throughout the proof of this lemma, we will call such stars *dangerous.* Note that stars that are not dangerous cannot make the partition unsafe.

Consider a dangerous star $star(a')$. Since the set $\mathcal{S}$ is $(\tilde{e}, j)$-partial, *either* it contains at most 2^{j-3} j-relevant copies of the vertex a' *or for all* vertices c in the star of a', the set $\mathcal{S}$ contains at most 2^{j-3} copies of the vertex c. In any case, with probability $1/2$, v belongs to the "right half" of M. (For example, if $\mathcal{S}$ contains at most $2^j/8$ copies of a', then $v \in M_{a'}$.) Since the number of dangerous stars is at most $8 \cdot \ln n$, it follows that

$$\text{Prob}\,(\text{safe partition}) \geq \left(\frac{1}{2}\right)^{8 \ln n} > \frac{1}{n^8},$$

proving the claim. $\square$

The following corollary is immediate

COROLLARY 4.6. *Let $\mathcal{S}$ be an $(\tilde{e}, j)$-partial set of size $|\mathcal{S}| \leq 2^j \ln n$. Then* $\mathbb{P}(\mathcal{S}\ \mathcal{AN}\ v) \geq 1/n^{12}$.

Proof. By Lemma 4.5, with probability $1/n^8$, the partitions of M_x are safe with respect to the set $\mathcal{S}$, i.e., for every vertex $x \in S_Y$, $s(x) \leq 2^{j-3}$. By Lemma 4.2,

$$\mathbb{P}(\mathcal{S}\ \mathcal{AN}\ v \mid \mathcal{S} \text{ safe partition}) \ \geq\ 1/n^4\ .$$

Hence, with probability at least $1/n^{12}$, the set $\mathcal{S}$ induces a safe partition, and the event $(\mathcal{S}\ \mathcal{AN}\ v)$ holds. □

4.2. Probability of a collision. In this section we consider sets $\mathcal{R} \subseteq \mathcal{V}_1$ of size at least 2^j that are analogous to R_i' in Lemma 3.2. In all the following sets, we are interested in the event that $\mathcal{R}$ does not inform v because it covers v at least twice (namely, v has at least two neighbors in $\mathcal{R}$). Let $(\mathcal{R}\ 2\mathcal{C}\ v)$ denote this event.

Clearly, if all the relevant events were independent, namely, if every vertex of $\mathcal{R}$ was connected to v with probability $1/2^j$ independently of other vertices, then

$$\mathbb{P}(\mathcal{R}\ 2\mathcal{C}\ v) \ =\ 1 - \left(1 - 1/2^j\right)^{|\mathcal{R}|} - \left(1 - 1/2^j\right)^{|\mathcal{R}|-1} \cdot |\mathcal{R}|. \tag{4.1}$$

This is in fact the case in the [1] construction.

We shall now see that in our construction, despite its dependencies, a similar inequality can be proven.

Consider a vertex x that contributes copies to $\mathcal{R}$, and suppose first that $v \notin M_x$. For such x, the edges between its copies and the vertex v are determined by the mixing step, and they behave exactly as in inequality (4.1). Similarly, if x_q is a copy of x and $x_q \in cp(x) \setminus cp(x,j)$, the probability that the edge (x, v) is in the graph is $1/2^j$ (see the mixing step).

However, the more delicate case is when $v \in M_x$. In this case the edges between the copies of x and the vertex v are determined by the random permutation step.

Let $X = \{x_1, \ldots, x_p\}$ be the set of copies of x in $\mathcal{R}$. First, we compute an upper bound on the probability of the event $(X\ \ \mathcal{AN}\ \ v)$, namely, that no vertex in X is connected to v by the random permutation step. Let $p' = \lceil p/2 \rceil$ and $X' = \{x_1, \ldots, x_{p'}\}$. Then

$$\begin{aligned}\mathbb{P}(X\ \ \mathcal{AN}\ \ v) \leq \mathbb{P}(X'\ \ \mathcal{AN}\ \ v) = \mathbb{P}(x_1\ \ \mathcal{AN}\ \ v) \cdot \mathbb{P}(x_2\ \ \mathcal{AN}\ \ v \mid x_1\ \ \mathcal{AN}\ \ v)&\\ \ldots \mathbb{P}(x_{p'}\ \ \mathcal{AN}\ \ v \mid \{x_1, \ldots, x_{p'-1}\}\ \ \mathcal{AN}\ \ v)&\ .\end{aligned}$$

If it is known that i of the copies of x are not connected to v by the random permutation step, then all the *rest* of the $2^j - i$ copies have equal probability of covering v. Thus, the probability that the next copy x_{i+1} is connected to v is

$$\frac{1}{2^j - i} \leq \frac{1}{2^{j-1}}.$$

This is because $i \leq p' - 1 \leq p/2 \leq 2^j/2$. Thus, we get that

$$\mathbb{P}(X\ \ \mathcal{AN}\ \ v) \leq \left(\frac{1}{2^{j-1}}\right)^{p'}.$$

In summary, the contribution of X to the probability is very similar to its contribution in inequality (4.1). The differences are $1/2^{j-1}$ instead of $1/2^j$ and p' (recall that

$p' \geq p/2$) instead of p. We derive an upper bound on the probability that v is informed by $\mathcal{R}$ in a similar way. Let $\rho = |\mathcal{R}|/2$. We have proved Lemma 4.7.

Lemma 4.7.

$$\mathbb{P}(\mathcal{R}\ 2\mathcal{C}\ v) \geq 1 - \left(1 - 1/2^{j-1}\right)^{\rho} + \rho \cdot \left(1 - 1/2^{j-1}\right)^{\rho - 1}. \qquad \square$$

4.3. Deriving a lemma similar to Lemma 3.2.

The pivot and the most significant index for $\tilde{e}$. For the rest of the section, consider a fixed short schedule $\Pi = (T_1, T_2, \ldots)$. We first define how to find the most "important" index j for $\tilde{e}$. Recall that $cp(A)$ (resp., $cp(B)$) is the set of all copies of vertices of A (resp., of vertices of B). Let $\Pi(\tilde{e})$ be the schedule $(R_1, R_2, \ldots)$ with $R_i = T_i \cap (cp(A) \cup cp(B))$. For the rest of the subsection, we use symbols R_i and $\mathcal{R}$ to denote rounds that are subsets of $cp(A) \cup cp(B)$. Recall that $|cp(A) \cup cp(B)| = n$ (because of the dummy vertices). Hence $cp(A) \cup cp(B)$ can play the role of the set $\mathcal{X}$ in Lemma 3.2.

Definition 4.8. *The index j, whose existence is guaranteed by Lemma 3.2 with respect to $\mathcal{X} = cp(A) \cup cp(B)$ and the rounds $\Pi(\tilde{e})$, is called the* pivot *of $\tilde{e}$.*

For the rest of the section we adopt a notation from the paper of Alon et al. [1]; let $\mathcal{S}$ be the set whose existence is guaranteed by Lemma 3.2 (i.e., it plays the role of S from Lemma 3.2), and $R'_i = R_i \setminus \mathcal{S}$.

The probability of 2-covering. The proof of the following lemma is very similar to the proof of Lemma 3.4 from [1]. The only difference between the two proofs is that in Lemma 3.2, the following inequality holds with respect to the schedule Π:

$$\mathbb{P}(\mathcal{R}\ 2\mathcal{C}\ v) \ \geq\ 1 - \left(1 - 1/2^{j}\right)^{|\mathcal{R}|} - \left(1 - 1/2^{j}\right)^{|\mathcal{R}|-1} \cdot |\mathcal{R}|.$$

This is because all the relevant events are independent. In our case, we use Lemma 4.7 which shows that, though the events are not independent, a similar inequality holds. We summarize as follows.

Lemma 4.9. *There is some universal constant c_1 so that*

$$\mathbb{P}(\textit{for all } i, R'_i\ 2\mathcal{C}\ v) \geq \frac{1}{n^{c_1}}. \qquad \square$$

How can we force $\mathcal{S}$ to be partial? In order to apply Corollary 4.6 to bound the probability that v is informed, we need the subset $\mathcal{S}$ (from Lemma 3.2) to be $\tilde{e}$-partial. How can we guarantee that? One way of ensuring this is by requiring that $\bigcup R_i$ is $\tilde{e}$-partial.

To understand our approach, assume for the moment that indeed $\bigcup R_i$ is $\tilde{e}$-partial. Then, we can derive a lemma similar to Lemma 3.2, that is, show that an $\tilde{e}$-partial schedule cannot cover all the vertices of $M_{\tilde{e}}$. We want to use this claim to get a good MIN-REP solution along the following lines:

1. With high probability, $\bigcup R_i$ cannot be *$\tilde{e}$-partial*, because of a lemma similar to Lemma 3.2.
2. This will hold in a similar way to many other superedges.
3. As $\bigcup R_i$ is not $\tilde{e}$-partial, there should exist an $\tilde{e}$-matching pair (x, y) so that $\bigcup R_i$ contains at least $2^j/8$ copies of x and at least $2^j/8$ copies of y. If this is the case, we say that Π *chose* x, y.
4. If $\mu = |\bigcup R_i|$ is "small," then $\mu/2^j$ is "small" as well. Hence Π can choose only a few matching pairs from $A \cup B$.
5. Hence, a small subset of $A \cup B$ can be used to cover $\tilde{e}$.

6. Since this applies to “many” superedges, we obtain a small solution for the original instance of the MIN-REP problem.

The problem with this scenario is in the case that $\mu = |\bigcup R_i|$ is “too large.” In this case $\mu/2^j$ can be very large by itself, and so the MIN-REP solution that will be derived may be large.

Indeed, it turns out that expecting that $\bigcup R_i$ is $\tilde{e}$-partial is too harsh a requirement, at least as far as very large rounds are present. The good news is, however, that large rounds have little effect because of Lemma 4.7. We next formalize this intuition. Again, we restrict our attention to $\Pi(\tilde{e})$ and consider rounds R that are subsets of $cp(A) \cup cp(B)$.

DEFINITION 4.10. *We say that a round R is $(\tilde{e}, j)$-small if*

$$|R| \leq c \cdot 2^j \ln n,$$

where c is some constant to be determined later. Let $Small(\Pi, j, \tilde{e})$ be the collection of small rounds of $\Pi(\tilde{e})$, and let $Large(\Pi, j, \tilde{e})$ be the set of all other rounds.

DEFINITION 4.11. *Let*

$$\mathcal{W}(\Pi, j, \tilde{e}) = \bigcup_{R_i \in Small(\Pi, j, \tilde{e})} R_i.$$

DEFINITION 4.12. *A schedule Π is $(\tilde{e}, j)$-partial if $\mathcal{W}(\Pi, j, \tilde{e})$ is $\tilde{e}$-partial.*

Note that even if Π is $\tilde{e}$-partial, still $\bigcup R_i$ may contain all the copies of *both* a and b for an $\tilde{e}$-matching pair (a, b). This is because some rounds R_i may be large.

Assume $c_0 \geq c_1 + 12$, where c_1 is the constant from Lemma 4.9. The following corollary (it is analogous to Lemma 3.2) is derived from Lemma 4.9 and Corollary 4.6. However, it applies only to $Small(\Pi, j, \tilde{e})$.

COROLLARY 4.13. *Let Π be $\tilde{e}$-partial. With probability at least $1/n^{c_0}$, $Small(\Pi, j, \tilde{e})$ does not inform v.*

Proof. We mimic the proof of Lemma 3.2. Namely, we compute the probability for the event $\mathcal{N}$ = “no vertex of $\mathcal{S}$ is connected to v” and the event $2\mathcal{C}$ = “all small rounds $R'_i = R_i \setminus \mathcal{S}$ cover v at least twice.” If both $\mathcal{N}$ and $2\mathcal{C}$ occur, then v is not informed by $Small(\Pi, j, \tilde{e})$.

Proving that the event “for every i, R'_i 2-covers v” occurs with probability at least $1/n^{c_1}$ is done exactly as in Lemma 4.9 and in [1].

Now we deal with $\mathcal{S}'$. As Π is $\tilde{e}$-partial, by definition $\mathcal{W}$ is $\tilde{e}$-partial, and thus $\mathcal{S}'$ is $\tilde{e}$-partial. Thus, from Corollary 4.6,

$$\mathbb{P}(\mathcal{S}' \ \ \mathcal{AN} \ \ v) \geq \frac{1}{n^{12}} \ .$$

Note that R'_i and $\mathcal{S}'$ are disjoint. But the events $\mathcal{N}$ and $2\mathcal{C}$ are *not* independent, as the two sets may contain copies of the same vertex. We need to study the correlation between $\mathcal{N}$ and $2\mathcal{C}$. We shall now see that the events are positively correlated.

Let R_x (resp., S_x) be the subset of copies of x that belong to R'_i (resp., $\mathcal{S}$). If $v \notin M_x$, then the edges between the vertices of R_x and S_x on the one hand and the vertex v on the other are defined by the mixing step and are completely independent. If $v \in M_x$, then the connection of $R_x \cup S_x$ and v is determined by the random permutation step. Now, if S_x is not connected to v, this only *increases* the probability that R_x is connected to v. Hence, the correlation between these probabilities is positive.

Therefore, with probability at least $\frac{1}{n^{c_1+12}} \geq \frac{1}{n^{c_0}}$, v is not informed by $Small(\Pi, j, \tilde{e})$. ∎

5. Analysis part II: Deriving the result.

Let Π be an $\tilde{e}$*-partial* short schedule.

5.1. How many vertices can $\boldsymbol{Small(\Pi, j, \tilde{e})}$ inform?.

We consider the number of vertices informed by $Small(\Pi, j, \tilde{e})$. (At this point we ignore the contribution of $Large(\Pi)$. We will deal with it later.)

Let q' be some index. We say that a set $M_{\tilde{e}}(j, q')$ is *fully informed* by $Small(\Pi, j, \tilde{e})$ if all the elements of $M_{\tilde{e}}(j, q')$ are informed. Let $NF = NF(\tilde{e}, j, Small(\Pi, j, \tilde{e}))$ be the number of indices q' for which $M_{\tilde{e}}(j, q')$ is *not* fully informed by $Small(\Pi, j, \tilde{e})$.

Lemma 5.1.

$$\mathbb{P}(NF < n^2) \leq \exp(-\Omega(n^3)).$$

Proof. Consider a fixed index q'. By Corollary 4.13 and the Markov inequality,

$$\mathbb{P}\left(M_{\tilde{e}}(j, q') \text{ is not fully informed}\right) \geq \frac{1}{n^{c_0 - 1}} .$$

Hence, the number of not fully informed sets $M_{\tilde{e}}(j, q')$ is a binomial variable with success probability greater than or equal to $1/n^{c_0-1}$. The number of different indices q' is n^{c_0+4}. Thus, the expected number of not fully informed $M_{\tilde{e}}(j, q')$ is at least n^3. Hence, the claim follows from the Chernoff bound. □

The lemma shows that $Small(\Pi, j, \tilde{e})$ not only does not inform all the vertices but also leaves *many* indices q' for which $M_{\tilde{e}}(j, q')$ has at least one noninformed element. In fact, a simple counting argument and the union-bound imply the following corollary.

Corollary 5.2. *With probability* $1 - \exp(-\Omega(n^3))$*, for* any $\tilde{e}$*-partial short schedule* Π*,* $NF(\tilde{e}, j, Small(\Pi, j, \tilde{e})) \geq n^2$.

Proof. The set of relevant vertices on a fixed round of Π is a subset of $cp(A) \cup cp(B)$. The size of $cp(A) \cup cp(B)$ is n. Hence, the number of subsets of $A \cup B$ is 2^n. Thus the number of short schedules is at most $2^{n \log^2 n}$. Since $2^{n \log^2 n} \ll exp(n^3)$, the claim follows from the union-bound. □

5.2. Large rounds are not able to inform many vertices.

By Corollary 5.2, with probability $1 - \exp(-\Omega(n^3))$, no short partial schedule satisfies $NF \leq n^2$. We next show that for any choice of Π, with high probability, $Large(\Pi)$ cannot "complete the task" and leaves some vertices uninformed.

Lemma 5.3. *If* Π *is* $\tilde{e}$*-partial short schedule, then with probability at least* $1 - 2^{-2n^2}$, Π *does not inform all the vertices of* $M_{\tilde{e}}(j)$.

Proof. We know that (with high probability) there is a set $\mathcal{U}$ of n^2 elements, $\mathcal{U} = \{u_1, \ldots, u_{n^2}\}$, that are not informed by $Small(\Pi, j, \tilde{e})$. The crucial property of $\mathcal{U}$ is that different vertices u_i belong to different sets $M_{\tilde{e}}(j, q')$. Hence the random events that we consider are independent. We fix the sets $Large(\Pi)$ and $\mathcal{U}$ and estimate the probability that $Large(\Pi)$ covers $M_{\tilde{e}}$.

Let $\mathcal{R} \subset A \cup B$ be a large round in $\Pi(\tilde{e})$. Let $r = \log_2 |\mathcal{R}|$. By definition, $\mu = |\mathcal{R}| \geq c \cdot 2^{j+1} \ln n$. By Lemma 4.7, setting

$$\rho = \frac{|\mathcal{R}|}{2},$$

we get

$$\mathbb{P}(\mathcal{R}\ 2\mathcal{C}\ v) \geq 1 - \left(1 - \frac{1}{2^{j-1}}\right)^{\rho} + \rho \cdot \left(1 - \frac{1}{2^{j-1}}\right)^{\rho - 1} .$$

Thus, it follows that the probability that some u_i is informed by $\mathcal{R}$ is at most $1/n^{c'}$ with c' being some universal constant (that depends on c from Definition 4.10). Since the edges between the vertices of $\mathcal{U}$ and v are independent (different u_i belong to different sets $M_{\tilde{e}}(j, q')$), the probability that the entire set $\mathcal{U}$ is informed is at most $1/n^{c'n^2}$.

Now, we count the number of possibilities to choose the sets $Large(\Pi)$ and $\mathcal{U}$. Note that $\mathcal{U}$ is a subset of size n^2 chosen out of a set of n^{c_0+4} vertices. The number of ways to do so is at most $n^{(c_0+4)n^2}$. The number of possible $Large(\Pi)$ schedules (restricted to subsets of $cp(A) \cup cp(B)$) is $O(2^{n \cdot \log^2 n})$. Thus the number of choices of $\mathcal{U}$ and $Large(\Pi)$ is at most $n^{(c_0+5)n^2}$. We set c so that $c' \geq c_0 + 7$. Now the claim follows from the union-bound. □

5.3. Short proper schedules cannot be feasible. We need the following definition. Intuitively, it defines short schedules that are partial for *many* superedges.

DEFINITION 5.4. *A short schedule* Π *is called* proper *if there exists a subset* $E' \subseteq \tilde{E}$ *that contains at least one-half of all the superedges and such that the* $Small(\Pi, j, \tilde{e})$ *is* $\tilde{e}'$*-partial for every* $\tilde{e}' \in E'$*. Otherwise, the schedule* Π *is called* nonproper*.*

The following lemma holds both for yes and no instances of the MIN-REP problem.

LEMMA 5.5. *With probability* $1 - \exp(\Omega(n^2))$*, no proper* Π *informs all the vertices of* $\mathcal{V}_2$*.*

Proof. First, fix E'. For a single superedge $\tilde{e}' \in E'$, the probability that $M_{\tilde{e}}$ is fully informed is at most 2^{-2n^2} (Lemma 5.3). Naturally, this also implies an upper bound on the probability that all the vertices of $\mathcal{V}_2$ are informed.

By restriction 2 in the definition of the MIN-REP problem, the number of superedges is bounded by $n \cdot \mathit{polylog}(n)$. Thus, the number of subsets of the superedges is at most $2^{o(n^2)}$. By the union-bound the probability that *there exists* a subset E' so that, for every $\tilde{e}' \in E'$, $M_{\tilde{e}'}$ is fully informed, is at most

$$2^{o(n^2)} \cdot 2^{-2n^2} = \exp(-\Omega(n^2)) . \qquad \square$$

Remark. Note that a schedule of logarithmic length for a yes instance that was described in section 3.7 is *not proper.*

5.4. For no instances short nonproper schedules are not feasible.

LEMMA 5.6. *With probability* 1 *there is no nonproper feasible short schedule* Π *for the instance derived out of a no instance of the MIN-REP problem.*

Proof. Suppose for contradiction that there exists a schedule Π as above. Assume, without loss of generality, that the first round of the schedule Π is the set $\{s\}$ and that all the other rounds $R \in \Pi$ are subsets of the set $\mathcal{V}_1$ (with no dummy vertices). Let $E_N \subseteq \tilde{E}$ be a subset of superedges such that, for every superedge $\tilde{e} \in E_N$, the schedule Π is not $\tilde{e}$-partial. By definition, the set E_N contains at least half of the superedges.

For a superedge $\tilde{e} \in E_N$, let j be its pivot. Recall that $\mathcal{W}$ is the union of all small rounds. By definition, $\mathcal{W}$ contains at least $2^j/8$ copies of both x and y for some $\tilde{e}$-matching pair (x, y). We call this pair "the important pair for $\tilde{e}$."

We now define a MIN-REP solution C that is both "of small size" and covers all the superedges of E_N. C is defined by the following procedure.

1. Go over all the superedges in E_N in an arbitrary order.
2. For a superedge $\tilde{e}$, let (x, y) be the important pair for $\tilde{e}$.
3. Add x and y to C.

The following claim is immediate by definition.

CLAIM 5.1. *C covers all the superedges E_N.*

We now bound $|C \cap A|$ for an arbitrary supervertex A.

Since the supervertex A participates in several different superedges, and each with its own pivot, we consider every index j separately. Fix some pivot j, and bound the contribution to $C \cap A$ due to j. Recall that the subschedule $Small(\Pi, j, \tilde{e})$ contains only the rounds R that are j-small with respect to the superedge $\tilde{e}$, i.e., the rounds $R \in \Pi(\tilde{e})$ that satisfy $|R| \leq 2^{j+1} \cdot c \cdot \ln n$. Also, the number of rounds in the schedule is $O(\log^2 n)$. Hence, $|\mathcal{W}| = O(2^j \cdot \log^3 n)$ (because $\mathcal{W}$ is the union of all small rounds.)

Every superedge (A, B) with pivot j causes the important $\tilde{e}$-pair (a, b) to be added into C. But, by definition, $\mathcal{W}$ contains at least $2^j/8$ copies of a and $2^j/8$ copies of b. In particular, the number of vertices a that can be added to C with pivot j is at most

$$\frac{|\mathcal{W}|}{2^j/8} = O(\log^3 n).$$

This bounds the contribution of j to $|A \cap C|$.

Summing over all different indices j, the total size of $A \cap C$ is bounded by $O(\log^4 n)$. Similar bound follows for every B.

In other words, we have shown that the set C covers at least one-half of all the superedges of the instance $\mathcal{M}$ of the MIN-REP problem and contains $O(\log^4 n)$ representative vertices from each supervertex. It follows that no A or B sets contribute more than $\log^{10} n$ vertices to C. This contradicts Theorem 2.5. $\square$

COROLLARY 5.7. *With high probability, for a no instance no short schedule Π is feasible.*

Proof. By Lemma 5.6, with probability 1, there is no nonproper feasible short schedule for the instance $\mathcal{G}$. By Lemma 5.5, with high probability there is no proper feasible short schedule for the instance $\mathcal{G}$. Since any schedule is either proper or nonproper, the assertion follows. $\square$

Hence, we have shown that the reduction has the claimed gap and have proved our main result.

THEOREM 5.8. *Unless $NP \subseteq BPTIME(n^{O(\log \log n)})$, for some universal constant c there is no additive $(c \cdot \log^2 n)$-approximation for the radio broadcast problem.*

Acknowledgment. The first-named author wishes to thank Oded Regev for helpful discussions.

REFERENCES

[1] N. ALON, A. BAR-NOY, N. LINIAL, AND D. PELEG, *A lower bound for radio broadcast*, J. Comput. System Sci., 43 (1991), pp. 290–298.

[2] S. ARORA, L. BABAI, J. STERN, AND Z. SWEEDYK, *The hardness of approximate optima in lattice codes*, in Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science, 1993, pp. 724–733.

[3] S. ARORA AND K. LUND, *Approximation Algorithms for NP-Hard Problems*, D. Hochbaum, ed., PWS Publishing, Boston, MA, 1996.

[4] R. BAR-YEHUDA, R. O. GOLDREICH, AND A. ITAI, *Efficient emulation of single-hop radio network with collision detection on multi-hop radio network with no collision detection*, Distributed Comput., 5 (1991), pp. 67–72.

[5] B. S. CHLEBUS, L. GASIENIEC, A. GIBBONS, A. PELC, AND W. RYTTER, *Deterministic broadcasting in ad hoc radio networks*, Distributed Comput., 15 (2002), pp. 27–38.

[6] M. CHROBAK, L. GASIENIEC, AND W. RYTTER, *Fast broadcasting and gossiping in radio networks*, in Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science, Redondo Beach, CA, 2000, pp. 575–581.

[7] I. Chlamtac and S. Kutten, *A spatial-reuse TDMA/FDMA for mobile multihop radio networks*, in Proceedings of the IEEE INFOCOM, 1985, pp. 389–394.

[8] A. F. Clementi, A. Monti, and R. Silvestri, *Distributed broadcast in radio networks of unknown topology*, Theoret. Comput. Sci., 302 (2003), pp. 337–364. Preliminary version appeared in Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms, Washington, DC, 2001, pp. 709–718.

[9] A. Czumaj and W. Rytter, *Broadcasting algorithms in radio networks with unknown topology*, in Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, 2003, pp. 492–501.

[10] I. Chlamtac and O. Weinstein, *The wave expansion approach to broadcasting in multihop radio networks*, in Proceedings of the IEEE INFOCOM, 1987, pp. 874–881.

[11] M. Elkin and G. Kortsarz, *A logarithmic lower bound for radio broadcast*, J. Algorithms, 52 (2004), pp. 8–25.

[12] M. Elkin and G. Kortsarz, *An improved algorithm for radio broadcast*, in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, Vancouver, British Columbia, Canada, 2005, pp. 222–231.

[13] N. Garg, G. Konjevod, and R. Ravi, *A polylogarithmic approximation algorithm for the group Steiner tree problem*, J. Algorithms, 37 (2000), pp. 66–84.

[14] I. Gaber and Y. Mansour, *Broadcast in radio networks*, in Proceedings of the 6th ACM-SIAM Symposium on Discrete Algorithms, 1995, pp. 577–585.

[15] R. Gandhi, S. Parthasarathy, and A. Mishra, *Minimizing broadcast latency and redundancy in ad hoc networks*, in Proceedings of the Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'03), 2003, pp. 222–232.

[16] E. Halperin and R. Krauthgamer, *Polylogarithmic inapproximability*, in Proceedings of the 35th Annual ACM Symposium on Theory of Computing, 2003, pp. 585–594.

[17] P. Indyk, *Explicit constructions of selectors and related combinatorial structures, with applications*, in Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms, 2002, pp. 697–704.

[18] E. Kushilevitz and Y. Mansour, *An $\Omega(D \log(n/D))$ lower bound for broadcast in radio networks*, SIAM J. Comput., 27 (1998), pp. 702–712.

[19] D. Kowalski and A. Pelc, *Deterministic broadcasting time in radio networks of unknown topology*, in Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002, pp. 63–72.

[20] D. Kowalski and A. Pelc, *Broadcasting in undirected ad hoc radio networks*, in Proceedings of the 16th ACM Symposium on Principles of Distributed Computing, 2003, pp. 73–82.

[21] D. Kowalski and A. Pelc, *Centralized deterministic broadcasting in undirected multi-hop radio networks*, in APPROX-RANDOM, Lecture Notes in Comput. Sci. 3122, Springer, Berlin, 2004, pp. 171–182.

[22] C. Lund and M. Yannakakis, *On the hardness of approximating minimization problems*, J. Assoc. Comput. Mach., 41 (1994), pp. 960–981.

[23] R. Raz, *A parallel repetition theorem*, SIAM J. Comput., 27 (1998), pp. 763–803.

SIAM J. DISCRETE MATH.
Vol. 19, No. 4, pp. 900–913

COMPUTING MINIMAL TRIANGULATIONS IN TIME $O(n^{\alpha} \log n) = o(n^{2.376})$*

PINAR HEGGERNES[†], JAN ARNE TELLE[†], AND YNGVE VILLANGER[†]

Abstract. The problem of computing minimal triangulations of graphs, also called minimal fill, was introduced and solved in 1976 by Rose, Tarjan, and Lueker [*SIAM J. Comput.*, 5 (1976), pp. 266–283] in time $O(nm)$ and thus $O(n^3)$ for dense graphs. Although the topic has received increasing attention since then and several new results on characterizing and computing minimal triangulations have been presented, this first time bound has remained the best. In this paper we introduce an $O(n^{\alpha} \log n)$ time algorithm for computing minimal triangulations, where $O(n^{\alpha})$ is the time required to multiply two $n \times n$ matrices. The current best known α is less than 2.376, and thus our result breaks the longstanding asymptotic time complexity bound for this problem. To achieve this result, we introduce and combine several techniques that are new to minimal triangulation algorithms, such as working on the complement of the input graph, graph search for a vertex set A that bounds the size of the connected components when A is removed, and matrix multiplication.

Key words. chordal graph, minimal triangulation, minimal fill, matrix multiplication

AMS subject classifications. 05C85, 68R10, 05C50

DOI. 10.1137/S0895480104445010

1. Introduction and motivation.

Any graph can be embedded in a chordal graph by adding a set of edges called fill, and the resulting graph is called a triangulation of the input graph. When the added set of fill edges is inclusion minimal, the resulting triangulation is called a minimal triangulation. The first algorithms for computing minimal triangulations were given in independent works of Rose, Tarjan, and Lueker [17] and Ohtsuki, Cheung, and Fujisawa [13, 14] in 1976. Among these, the algorithms of [13] and [17] have a time bound of $O(nm)$, where n is the number of vertices and m is the number of edges of the input graph. These first algorithms were motivated by the need to find good pivotal orderings for Gaussian elimination, and the mentioned papers gave characterizations of minimal triangulations through minimal elimination orderings. Since then, the problem has received increasing attention, and several new characterizations of minimal triangulations connected to minimal separators of the input graph have been given [5, 10, 15], totally independent of the connection to Gaussian elimination. The connection to minimal separators has increased the importance of minimal triangulations from a graph theoretical point of view, and minimal triangulations have proved useful in reconstructing evolutionary history through phylogenetic trees [9]. As a result, algorithms based on the new characterizations have been given [3, 8], while at the same time new algorithms based on elimination orderings also have appeared [4, 7, 16]. However, the best time bound remained unchanged, and trying to break the asymptotic $O(n^3)$ bound of computing minimal triangulations, in particular for dense graphs, became a major theoretical challenge concerning this topic.

In this paper, we introduce an $O(n^{\alpha} \log n)$ time algorithm to compute minimal

*Received by the editors July 7, 2004; accepted for publication (in revised form) June 15, 2005; published electronically December 30, 2005. A preliminary version of this paper appeared at the 16th ACM–SIAM Symposium on Discrete Algorithms (SODA), 2005.
http://www.siam.org/journals/sidma/19-4/44501.html

[†] Department of Informatics, University of Bergen, N-5020 Bergen, Norway (pinar@ii.uib.no, telle@ii.uib.no, yngvev@ii.uib.no).

triangulations of arbitrary graphs, where $O(n^\alpha)$ is the time bound of multiplying two $n \times n$ matrices. Currently the lowest value of α is $2.375 < \alpha < 2.376$ by the algorithm of Coppersmith and Winograd [6]. Hence the current time bound for our algorithm is $o(n^{2.376})$ since $\log n = o(n^\epsilon)$ for all $\epsilon > 0$. In order to achieve this time bound, we use several different techniques, one of which is matrix multiplication, to make parts of the input graph into cliques. Our algorithm runs for $O(\log n)$ iterations, and at each iteration the total work is bounded by the time needed for matrix multiplication. In order to achieve $O(\log n)$ iterations, we show how to recursively divide the problem into independent subproblems of a constant factor smaller size using a specialized search technique. In order to bound the amount of work at each iteration by $O(n^\alpha)$, we store and work on the complement graphs for each subproblem, in which case the subproblems do not overlap in any (non)edges. In addition, we use both the minimal separators and the potential maximal cliques of the input graph, combining the results of [5], [10], and [15].

Independent of our work, a very recent and thus yet unpublished result of Kratsch and Spinrad [12] uses matrix multiplication to give a new implementation of the minimal triangulation algorithm Lex M from 1976 [17]. Based on the matrix multiplication algorithm of [6], their presented time complexity is $O(n^{2.688})$. Other than the use of matrix multiplication, their approach is totally different from ours. Kratsch and Spinrad used matrix multiplication for similar problems in their SODA 2003 paper [11].

After the next section which contains some basic definitions, we give the main structure of our algorithm in section 3, followed by the important subroutine for partitioning into balanced subproblems in section 4. We tie these parts together in the last section.

2. Background and notation.

We consider simple undirected and connected graphs $G = (V, E)$ with $n = |V|$ and $m = |E|$. When G is given, denote the vertex and edge set of G by $V(G)$ and $E(G)$, respectively. For a set $A \subseteq V$, $G(A)$ denotes the subgraph of G induced by the vertices in A. A is called a *clique* if $G(A)$ is complete. The process of adding edges to G between the vertices of $A \subseteq V$ so that A becomes a clique in the resulting graph is called *saturating* A. The *neighborhood* of a vertex v in G is $N_G(v) = \{u \mid uv \in E\}$, and the *closed neighborhood* of v is $N_G[v] = N_G(v) \cup \{v\}$. Similarly, for a set $A \subseteq V$, $N_G(A) = \cup_{v \in A} N_G(v) \setminus A$, and $N_G[A] = N_G(A) \cup A$. $|N_G(v)|$ is the *degree* of v. When graph G is clear from the context, we will omit subscript G.

A vertex set $S \subset V$ is a *separator* if $G(V \setminus S)$ is disconnected. Given two vertices u and v, S is a *u, v-separator* if u and v belong to different connected components of $G(V \setminus S)$, and S is then said to *separate* u and v. Two separators S and T are said to be *crossing* if S is a u, v-separator for a pair of vertices $u, v \in T$, in which case T is an x, y-separator for a pair of vertices $x, y \in S$ [10, 15]. A u, v-separator S is *minimal* if no proper subset of S separates u and v. In general, S is a *minimal separator* of G if there exist two vertices u and v in G such that S is a minimal u, v-separator. It can be easily verified that S is a minimal separator if and only if $G(V \setminus S)$ has two distinct connected components C_1 and C_2 such that $N_G(C_1) = N_G(C_2) = S$. In this case, C_1 and C_2 are called *full components*, and S is a minimal u, v-separator for *every* pair of vertices $u \in C_1$ and $v \in C_2$.

A *chord* of a cycle is an edge connecting two nonconsecutive vertices of the cycle. A graph is *chordal*, or equivalently *triangulated*, if it contains no chordless cycle of length ≥ 4. A graph $G' = (V, E \cup F)$ is called a *triangulation* of $G = (V, E)$ if G' is chordal. The edges in F are called *fill edges*. G' is a *minimal triangulation* if

$(V, E \cup F')$ is nonchordal for every proper subset F' of F. It was shown in [17] that a triangulation G' is minimal if and only if every fill edge is the unique chord of a 4-cycle in G'. Another characterization of minimal triangulations which is central to our results is that G' is a minimal triangulation of G if and only if G' is the result of saturating a maximal set of pairwise noncrossing minimal separators of G [15].

By the results of Kloks, Kratsch, and Spinrad [10] and Parra and Scheffler [15], it can be shown that the following recursive procedure creates a minimal triangulation of G: Take any connected vertex subset K, and let $A = N[K]$; compute the connected components $C_1, \ldots, C_k$ of $G(V \setminus A)$; saturate each set $N(C_i)$ for $1 \leq i \leq k$, and call the resulting graph G'; and then compute a minimal triangulation of each subgraph $G'(N[C_i]), 1 \leq i \leq k$, and of $G'(A)$ independently. The key to understanding this is to note that the saturated sets $N(C_i)$ are noncrossing minimal separators of G and G'. Thus the problem decomposes into independent subproblems overlapping only at the saturated minimal separators, and we can continue recursively on each subproblem that is not complete. This procedure is basic to the main structure of our algorithm.

An extension of the above mentioned results, which we also use in our algorithm, was presented by Bouchitté and Todinca in [5]. There, a *potential maximal clique (pmc)* of G is defined to be a maximal clique in some minimal triangulation of G. If A is a pmc, then it is shown in [5] that whole A will automatically be saturated in the above recursive procedure instead of appearing as a subproblem and that this modified procedure indeed characterizes minimal triangulations. In this case A is not necessarily $N[K]$ for a connected set K. The following theorem from [5] characterizes a pmc, and it will be used to prove the correctness of our balanced partition algorithm in section 4.

THEOREM 2.1 (Bouchitté and Todinca [5]). *Given a graph $G = (V, E)$, let $P \subseteq V$ be any set of vertices, and let $C_1, C_2, \ldots, C_k$ be the connected components of $G(V \setminus P)$. P is a pmc of G if and only if the following hold:*

1. *$G(V \setminus P)$ has no full component, and*
2. *P is a clique when every $N(C_i)$ is saturated for $1 \leq i \leq k$.*

3. The new algorithm and the data structures.

Observe that the total work for saturating all sets $N(C_i), 1 \leq i \leq k$, in the recursive procedure described in the previous section requires $O(n^3)$ time if it is done straightforwardly, as these sets might overlap heavily and contain $O(n)$ vertices each. With the help of matrix multiplication, this total time can be reduced to $O(n^\alpha)$. We construct the following matrix $M = M_{G,A}$: for each vertex $v \in V(G)$ there is a row in M, for each connected component C of $G(V \setminus A)$ there is a column in M, and entry $M(v, C) = 1$ if $v \in N(C)$. All other entries are zero. Now we perform the multiplication MM^T, and in the resulting symmetric matrix, entry $(u, v) = (v, u)$ is nonzero if and only if u and v both belong to a common set $N(C)$ for some C. Thus MM^T is the adjacency matrix of a graph in which each $N(C)$ is a clique. The use of matrix multiplication for this purpose was first mentioned in [11].

Once MM^T is computed, the edges indicated by its nonzero entries can be added to G, resulting in the partially filled graph G', and the subproblems $G'(N[C_i]), 1 \leq i \leq k$, and $G'(A)$ can be extracted. Now for each subproblem this process can be repeated recursively. However, it is important that we do not perform a matrix multiplication for each subproblem in the further process but that we create only *one* matrix and perform a single matrix multiplication for all subproblems of each level in the recursion tree. Thus in the resulting matrix MM^T, entry (u, v) is nonzero if and only if there is a connected component C of one of the subproblems of this level such

```
Algorithm FMT - Fast Minimal Triangulation
Input: An arbitrary noncomplete graph $G = (V, E)$.
Output: A minimal triangulation $G'$ of $G$.

Let $Q_1, Q_2$ and $Q_3$ be empty queues;   Insert $G$ into $Q_1$;
$G' = G$;
repeat
    Construct a zero matrix $M$ with a row for each vertex in $V$ (columns are added
    later);
    while $Q_1$ is nonempty do
        Pop a graph $H = (U, D)$ from $Q_1$;
        Call Algorithm Partition($H$) which returns a vertex subset $A \subset U$;
        Push vertex set $A$ onto $Q_3$;
        for each connected component $C$ of $H(U \setminus A)$ do
            Add a column in $M$ such that $M(v, C) = 1$ for all vertices $v \in N_H(C)$;
            if $\exists$ nonedge $uv$ in $H(N_H[C])$ with $u \in C$ then
                Push $H_C = (N_H[C], D_C)$ onto $Q_2$, where $uv \notin D_C$ only if $u \in C$ and
                $uv \notin D$; [1]
        end-for
    end-while
    Compute $MM^T$;
    Add to $G'$ the edges indicated by the nonzero elements of $MM^T$;
    while $Q_3$ is nonempty do
        Pop a vertex set $A$ from $Q_3$;
        if $G'(A)$ is not complete then Push $G'(A)$ onto $Q_2$;
    end-while
    Swap names of $Q_1$ and $Q_2$;
until $Q_1$ is empty
```

FIG. 1. *Algorithm FMT: Fast Minimal Triangulation.*

that $u, v \in N_{G'}(C)$. For this reason, we cannot actually use recursion, and we have to keep track of all subproblems belonging to the same level. We do this by using two queues Q_1 and Q_2 which will memorize all subproblems for the current and next level, respectively. Only those new subproblems that are not cliques in the partially filled graph should survive to the next iteration. For a new subproblem on vertex set $N[C_i]$ appearing from a connected component C_i, after removing A we check this before the saturation as we already know that the saturation will make $N(C_i)$ into a clique and not add any other edges to the graph induced by $N[C_i]$. However, for the subproblem on vertex set A itself we must wait until after the saturation before checking whether A now induces a clique, and for that reason we store the vertex sets A temporarily in a third queue Q_3.

Our algorithm, which we call FMT, for fast minimal triangulation, is given in Figure 1. The process of computing a good vertex set A is the most complicated part of this algorithm, and this part will be explained in the next section when we give the details of Algorithm Partition that returns such a set A. For the time being, and for the correctness of Algorithm FMT, it is important and sufficient to note that Algorithm Partition returns a set A, where either $A = N[K]$ for some connected vertex set K or A is a pmc.

The following lemma proves the correctness of our algorithm as well as the correctness of the recursive procedure described in the previous section.

LEMMA 3.1. *Algorithm FMT computes a minimal triangulation of the input graph as long as the Partition(H) subroutine returns a set $A \subset V(H)$, where either $A = N[K]$ for some connected vertex set K or A is a pmc.*[1]

Proof. Let $G = (V, E)$ be the input graph, and let K be a set of vertices such that $G(K)$ is connected. It is shown in [1] that the set of minimal separators of G that are subsets of $N(K)$ is exactly the set $\{N(C) \mid C$ is a connected component of $G(V \setminus N[K])\}$. In [5] it is shown that if P is a pmc, then the set of minimal separators that are contained in P is exactly the set $\{N(C) \mid C$ is a connected component of $G(V \setminus P)\}$.

Since A is always chosen so that either $A = N[K]$ for a connected set K or A is a pmc (this will be proved in section 4), then it follows that all sets that are saturated at the first iteration of Algorithm FMT are minimal separators of G. We will now argue that these minimal separators are noncrossing. Assume on the contrary that two crossing separators $S = N(C_1)$ and $T = N(C_2)$ are saturated at the first iteration, where C_1 and C_2 are two distinct connected components of $G(V \setminus A)$. Thus there are two vertices $u, v \in T$ with $u, v \notin S$ such that S is a minimal u, v-separator in G. Since $u, v \in T = N(C_2)$, and S does not contain any vertex of C_2, the removal of S cannot separate u and v as there is a path between u and v through vertices of C_2. This contradicts the assumption that S is a u, v-separator, and thus we can conclude that the minimal separators saturated at the first step are all pairwise noncrossing. It is important to observe that once these separators are saturated, all minimal separators of G that cross any of these will disappear as the saturated sets do not contain pairs of vertices that are separable. At each iteration, any minimal separator of G' is a minimal separator of G [15]. Thus the minimal separators that we discover at each iteration will not cross the minimal separators discovered and saturated at previous iterations.

At each new iteration, the above argument can be applied to each subgraph H, and thus we compute a set of noncrossing minimal separators of each subgraph H at each iteration. We have already argued that these cannot cross any of the saturated minimal separators of previous iterations. We must also argue that no minimal separator of a subgraph of an iteration crosses a minimal separator of another subgraph of the same iteration. But this is straightforward as these subgraphs only intersect at cliques, and thus their sets of minimal separators are disjoint.

So, our algorithm computes and saturates a set of noncrossing minimal separators at each iteration. Since we continue this process until all minimal separators of G' are saturated, by the results of [10] and [15], we create a minimal triangulation. $\square$

If we consider merely correctness, any set A that fulfills the requirements can be chosen arbitrarily; for example, $A = N[u]$ for a single vertex u, as in [2]. In order to achieve the desired time complexity, we will devote the next section to describing how to carefully choose a vertex subset A in each subproblem so that the number of iterations of the repeat loop becomes $O(\log n)$.

In this section, we will argue that each iteration of the algorithm can be carried out in $O(n^\alpha)$ time. We start with the following lemma, which will give us the desired bound for the matrix multiplication step.

LEMMA 3.2. *At each iteration of Algorithm FMT, the number of columns in*

[1] What we want to do here is to take $H(N_H[C])$, make $N_H(C)$ into a clique, and then insert the resulting graph into Q_2. However, we do not have time to even compute $H(N_H[C])$. Thus we start with a complete graph on vertex set $N_H[C]$ and remove only edges uv with an endpoint in u that do not appear in D.

matrix M is less than n.

Proof. The sequence of iterations of the algorithm gives rise to an iterative refinement of a tree-decomposition of the graph G', a property first shown for the LB-treedec algorithm discussed in [8]. Simplifying the standard notation, we say that a *tree-decomposition* T_i of a graph G is a collection of *bags*, subsets of the vertex set of G, arranged as nodes of a tree such that the bags containing any given vertex induce a connected subtree and such that every pair of adjacent vertices of G is contained in some bag (see, e.g., page 549 of [19] for the standard definition). At the first iteration we have the trivial tree-decomposition T_1 with all vertices of G' in a single bag until the last iteration p, where the tree-decomposition T_p is in fact a clique tree of the now chordal graph G', with each bag inducing a unique maximal clique. We prove this by showing the following.

Loop invariant. At the start of iteration s we have a tree-decomposition T_s of the current partially filled graph G' whose bags consist of some vertex subsets inducing cliques, which are the vertices of subproblems inducing cliques as discovered so far by our algorithm, and where remaining bags are the vertex sets of subproblems in Q_1. The intersection of two neighboring bags in T_s is a saturated minimal separator of G' and thus induces a clique. T_s is nonredundant, meaning that if A, B are bags of T_s, then we do not have $A \subseteq B$.

The invariant is clearly true for the trivial tree-decomposition T_1 with a single bag. Let vertex set U be a bag of T_s appearing as subproblem $H = (U, D)$ in Q_1. The algorithm proceeds to find $A \subset U$ and produces new vertex subsets $A, N[C_1], N[C_2], \ldots, N[C_k]$, where each C_i is a component of $G'(U \setminus A)$. The node of bag U in T_s is in T_{s+1} split into a k-star with center-bag A and leaf-bags $N[C_1]$, $N[C_2], \ldots, N[C_k]$. Since A is a pmc or $A = N[K]$, it follows that this star is a tree-decomposition of $G'(U)$ which is nonredundant. The node of a neighboring bag X of U in the tree of T_s will also be split into a star unless X induces a clique, in which case it remains a single node, i.e., a trivial star. These two stars appearing from adjacent nodes in T_s will be joined in T_{s+1} by an edge between two bags U' and X' that each contain $U \cap X$. Such a bag must exist in each star since $U \cap X$ already induced a clique.

The tree-decomposition T_{s+1} is constructed by applying the construction above to each bag, and to adjacent pairs of bags, of T_s. After newly found minimal separators in G' have been saturated, then T_{s+1} will be a tree-decomposition of G', as is easily checked. It remains to show that T_{s+1} is nonredundant. We do this by showing that none of the new vertex subsets $A, N[C_1], N[C_2], \ldots, N[C_k]$ are contained in $U \cap X$. The crucial fact is that each vertex in $U \cap X$ has a neighbor in $U \setminus X$ since $U \setminus X$ was a component of the minimal separator $U \cap X$. If A was chosen as $A = N[K]$, then even if $K \subseteq U \cap X$, we would not have $A \subseteq U \cap X$. Likewise, we could have some component C_i of $G'(U) \setminus A$ with $C_i \subseteq U \cap X$, but we would never have $N[C_i] \subseteq U \cap X$. If A instead was chosen as a pmc, then we cannot have $A \subseteq U \cap X$, as $U \cap X$ was a minimal separator and a maximal clique cannot be part of a minimal separator. Thus, T_{s+1} is nonredundant. Since any bag of T_{s+1} that does not induce a clique is put back onto Q_1 before the next iteration, we have established the loop invariant.

Note that each column added to matrix M in the algorithm gives rise to a unique bag of T_{s+1}. Since the number of bags in the final tree-decomposition T_p is at most n, one for each maximal clique in a chordal graph, and since the number of bags in trees $T_1, \ldots, T_p$ is strictly increasing, we have proved the lemma. □

Consequently, the matrix multiplication step requires $O(n^\alpha)$. In order to be able to bound the time for the rest of the operations of each iteration by $O(n^\alpha)$, we will store and work on the *nonedges*, i.e., the edges of the complement graph for each subproblem. Note that subproblems can overlap both in vertices and in edges, which makes it difficult to bound the sum of their sizes for the desired time analysis. A nonedge uv is discarded when it becomes an edge (that is, when it is added to the graph) or when vertices u and v are separated into different subproblems, and if it is not discarded, it appears only in a single subproblem in the next iteration. Hence subproblems overlap only in cliques, so if we work on the complement of these subgraphs, then they actually do not overlap in any edges at all!

For each subgraph $H = (U, D)$ in Q_1, let $\bar{E}(H) = \binom{U}{2} \setminus D$ be the set of nonedges of H. Our data structure for each subproblem H is the adjacency list of $\bar{H} = (U, \bar{E}(H))$, where we also store the degree of each vertex in $\bar{H}$. It is an easy exercise to show that all linear time operations that we need to do for H, like computing the connected components and neighborhoods, can be done using only $\bar{H}$ in time $O(|\bar{E}(H)|+|V(H)|)$.

An interesting point is also that, when complement graphs are used, matrix multiplication is not necessary to saturate $N_H(C)$ of each subproblem $N_H[C]$; however, it is still necessary in order to saturate the subsets of A that become cliques. In the implementation of our algorithm, for each subproblem $H(N_H[C])$, we push the complement graph consisting of all nonedges of $H(N_H[C])$ with at least one endpoint in C onto Q_2. (This corresponds to Line 12 of Algorithm FMT.) We do this only if such a nonedge of H exists. Since these complement graphs consist of nonedges of $H(C)$ and nonedges of $H(N_H[C])$ between C and $N_H(C)$, all such subproblems can be computed in a total time of $O(|\bar{E}(H)| + |V(H)|)$ for H. Since we omit all nonedges between vertices belonging to $N_H(C)$, this actually corresponds to saturating $N_H(C)$ automatically.

After the matrix multiplication step, we look up in MM^T every edge of the complement of G' to check whether or not this nonedge should survive or should be deleted because it has now become a fill edge of G'. Since subproblems do not overlap in any nonedges, checking whether or not $G'(A)$ is now complete can be done in a total of $O(n^2)$ time for all vertex subsets A in Q_3.

Thus, for the implementation of our algorithm, we compute $\bar{G}$ at the beginning and use the complement graphs throughout the algorithm. Therefore, all operations described within an iteration can be completed within $O(n^\alpha)$ time. For clarity, we will give the algorithms on the actual graphs and not on complement graphs. We denote the set of nonedges of graph H by $\bar{E}(H)$.

With the given data structures and explanations, it should be clear that all operations during one iteration, outside of Algorithm Partition, can be performed in $O(n^\alpha)$ time.

4. Efficient partition into balanced subproblems.

In this section we will show how to compute vertex subsets A for each subproblem H in order to achieve an even partitioning into subproblems. Since each subproblem that results from H will not contain more than $\frac{4}{5}|\bar{E}(H)|$ nonedges, this will guarantee $O(\log n)$ iterations of the while-loop of Algorithm FMT.[2] The algorithm that we present for doing this will have running time $O(|\bar{E}(H)| + |V(H)|)$ on each input subgraph H.

The computation of vertex subset A for each subgraph $H = (U, D)$ is done by Algorithm Partition, which is given in Figure 2. This algorithm examines every vertex of H and tries to place it into a connected component C that results from removing

[2] The constant $\frac{4}{5}$ can in fact be replaced by $\frac{q-1}{q}$ for any $q \geq 5$. An implementation could make use of this fact to experimentally find the best value of q.

```
Algorithm Partition
Input:    A graph $H = (U, D)$ (a subproblem popped from $Q_1$).
Output:   A subset $A \subset U$ such that either $A = N[K]$ for some connected $H(K)$
          or $A$ is a pmc of $H$ (and $G'$).

Part I: defining P
Unmark all vertices of $H$;
$k = 1$;
while $\exists$ unmarked vertex $u$ do
    if $\mathcal{E}_{\bar{H}}(U \setminus N_H[u]) < \frac{2}{5}|\bar{E}(H)|$ then
        Mark $u$ as an s-vertex (stop vertex);
    else
        $C_k = \{u\}$;
        Mark $u$ as a c-vertex (component vertex);
        while $\exists\ v \in N_H(C_k)$ which is unmarked or marked as an s-vertex do
            if $\mathcal{E}_{\bar{H}}(U \setminus N_H[C_k \cup \{v\}]) \geq \frac{2}{5}|\bar{E}(H)|$ then
                $C_k = C_k \cup \{v\}$;
                Mark $v$ as a c-vertex (component vertex);
            else
                Mark $v$ as a p-vertex (pmc vertex);
                Associate $v$ with $C_k$;
            end-if
        end-while
        $k = k + 1$;
    end-if
end-while
$P$ = the set of all p-vertices and s-vertices;

Part II: defining A
if $H(U \setminus P)$ has a full component $C$ then
    $A = N_H[C]$;
else if there exist two nonadjacent vertices $u, v$ such that $u$ is an s-vertex
and $v$ is an s-vertex or a p-vertex then
    $A = N_H[u]$;
else if there exist two nonadjacent p-vertices $u$ and $v$, where $u$ is associated with $C_i$
and $v$ is associated with $C_j$ and $u \notin N_H(C_j)$ and $v \notin N_H(C_i)$ then
    $A = N_H[C_i \cup \{u\}]$;
else
    $A = P$;
end-if
```

FIG. 2. *Algorithm Partition.*

some set P of vertices from H, as long as $H(N_H[C])$ does not become too large with respect to the number of nonedges. The vertices that cannot be placed into any C with a small enough $H(N_H[C])$ in this way constitute exactly the set P whose removal from H results in these balanced connected components. Using this method, we compute a vertex set P such that all connected components C of $H(U \setminus P)$ have the nice property that $H(N_H[C])$ contains less than a constant factor of the nonedges

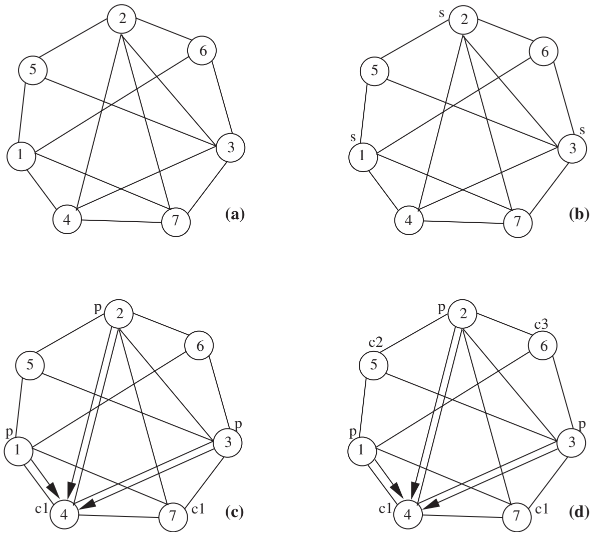


FIG. 3. *We give an example of how set P is found from graph H. In* (a)*, the number of nonedges $|\bar{E}(H)| = 7$, and the important bound for finding P is therefore $\frac{2}{5}|\bar{E}(H)| = 2.8$. First, the algorithm decides if vertex* 1 *can be contained in component C_1 by performing the test $\mathcal{E}_{\bar{H}}(U \setminus N_H[1]) < \frac{2}{5}|\bar{E}(H)|$. Vertex set $U \setminus N_H[1] = \{2, 3\}$, and $\mathcal{E}_{\bar{H}}(\{2, 3\}) = |N_{\bar{H}}(2)| + |N_{\bar{H}}(3)| = 2$. Thus* 1 *cannot be contained in a component, and it is marked as an* **s***-vertex. The same result is obtained when testing* 2 *and* 3*, as shown in* (b)*. Vertex set $U \setminus N_H[4] = \{5, 6\}$, and $\mathcal{E}_{\bar{H}}(\{5, 6\}) = |N_{\bar{H}}(5)| + |N_{\bar{H}}(6)| = 6 > 2.8$. Thus vertex* 4 *becomes the first vertex in component C_1. The algorithm will now try to extend C_1 by including vertices from $N(C_1)$ in C_1. Observe that $1 \in N(C_1)$ and that including it in C_1 will make the value of the test $\mathcal{E}_{\bar{H}}(U \setminus N_H[C_1 \cup \{1\}]) \geq \frac{2}{5}|\bar{E}(H)|$ false, and thus* 1 *becomes a* **p***-vertex and is associated to C_1 as shown in* (c)*. The same argument is used to change the marks of* 2 *and* 3 *as* **p***-vertices and to associate these with C_1. For vertex* 7 *we get the opposite result from the test, and therefore this vertex is placed in C_1.* (c) *shows that* 4 *and* 7 *are marked as* **c***-vertices, and the index after the* **c** *indicates that they belong to C_1. Finally, in* (d) *we create the components C_2 and C_3 containing vertices* 5 *and* 6*, respectively. All vertices in the neighborhood of these components are already marked as* **p***-vertices, and thus there is nothing more to do. As a result, the computed set $P = \{1, 2, 3\}$. For the rest of Algorithm Partition, since each connected component of $H(U \setminus P)$ is a full component, case* 1 *will apply, and the resulting returned set A is simply the union of P with one of these components, for example, $A = \{1, 2, 3, 6\}$. Note that there exist extreme cases where every vertex is marked as an* **s***-vertex. An example of this is a cycle of length* 16 *with added chords so that every vertex is adjacent to all vertices except the one on the opposite side of the cycle. The number of nonedges in this graph is* 8*, and the graph induced by any vertex and its neighborhood contains* 7 *nonedges. Such an extreme case causes no problem for our algorithm as case* 2 *will apply and an appropriate $A \subset U$ will still be found.*

of H. The computation of P is illustrated by an example given in Figure 3.

However, after P is computed, we cannot bound the number of nonedges that will belong to $G'(P)$ after the saturation. Furthermore, it might be the case that neither $P = N_H[K]$ for a connected vertex set K as required, nor P is a potential maximal clique, which implies that $N(C)$ is not necessarily a minimal separator for every connected component C of $H(U \setminus P)$. Thus we cannot simply use P as our desired set A. The set A is instead obtained using information gained through the computation of P, and we prove in Theorem 4.3 that it fulfills the requirements that were used to prove the correctness of Algorithm FMT and that the resulting subproblems all have at most $\frac{4}{5}|\bar{E}(H)|$ nonedges.

During Algorithm Partition, the vertices that we are able to place into small enough connected components are marked as **c**-vertices. The remaining vertices (which constitute P) are of two types: **p**-vertices have neighbors in a connected component of $H(U \setminus P)$, whereas **s**-vertices do not. For each connected component C of $H(U \setminus P)$ we want to ensure that the number $|\bar{E}(H(N_H[C]))|$, i.e., the number of nonedges with both endpoints in $N_H[C]$, is less than some fraction of $|\bar{E}(H)|$. The obstacle is that we cannot compute this number straightforwardly for all connected components of $H(U \setminus P)$ in the given time since the nonedges between vertices in $P \cap N_H[C]$ could be contained in too many such computations. However, we are able to give upper and lower bounds on $|\bar{E}(H(N_H[C]))|$ by summing the degrees in $\bar{H}$ of vertices in each $N_H[C]$, which we compute in the following roundabout manner in order to stay within the time limits. Define $\mathcal{E}_{\bar{H}}(S)$ to be the sum of degrees in $\bar{H}$ of vertices in $S \subseteq U = V(H)$. Since the sum of degrees is equal to twice the number of edges, we have $\mathcal{E}_{\bar{H}}(S) = 2|\bar{E}(H)| - \mathcal{E}_{\bar{H}}(U \setminus S)$. The quantity $\mathcal{E}_{\bar{H}}(U \setminus N_H[C])$ we indeed do have the time to compute, as we will explain in the proof of Lemma 4.1:

$$\mathcal{E}_{\bar{H}}(U \setminus N_H[C]) = \sum_{v \in U \setminus N[C]} |N_{\bar{H}}(v)|.$$

When checking whether $\mathcal{E}_{\bar{H}}(U \setminus N_H[C_k \cup \{v\}]) \geq \frac{2}{5}|\bar{E}(H)|$ in Algorithm Partition, we are indirectly checking whether $|\bar{E}(N_H[C_k \cup \{v\}])| \leq \frac{4}{5}|\bar{E}(H)|$, which is what we indeed want to know. The discussion in the proof of Lemma 4.2 explains this connection. The value $\mathcal{E}_{\bar{H}}(U \setminus N_H[C_k \cup \{v\}])$ can be computed in $O(|N_{\bar{H}}(v)|)$ time for each vertex v in U, as we show in the proof of the following lemma.

LEMMA 4.1. *Running Algorithm Partition on all subgraphs H of a single iteration of Algorithm FMT requires a total of $O(n^2)$ time.*

Proof. First we prove that the running time of Algorithm Partition on input subgraph H is $O(|\bar{E}(H)| + |V(H)|)$, and then we will argue for the overall time bound at the end. Note that, as explained in the previous section, also for Algorithm Partition we will work on the complement graph $\bar{H}$ for an efficient implementation. Observe that between a connected component C and $U \setminus N_H[C]$, we have a complete bipartite graph in $\bar{H}$, meaning that no vertex of C is adjacent to any vertex of $U \setminus N_H[C]$ in H. These nonedges will be used as an argument to obtain the desired time bound.

The pseudocode of Algorithm Partition is presented in two bulks. Let us call the first bulk "defining P" and the second bulk "defining A."

The first operation in the "defining P" part is to unmark every vertex in H. The value $\mathcal{E}_{\bar{H}}(U \setminus N_H[u])$ for a single vertex u is computed straightforwardly by summing the degrees in the complement graph of all vertices in $U \setminus N_H[u] = N_{\bar{H}}(u)$, which is an $O(|N_{\bar{H}}(u)|)$ operation.

When a component C_k is created from a first vertex u, we label every vertex $w \in N_{\bar{H}}(u)$ with the value $nk + |C_k| = nk + 1$. By labeling the vertices in this way, we assign a unique value to every vertex set that constitutes a component during the algorithm and ensure that only vertices in $U \setminus N_H[C_k]$ can have the label $nk + |C_k|$. The value $\mathcal{E}_{\bar{H}}(U \setminus N_H[C_k \cup \{v\}])$ can now be computed in $O(|N_{\bar{H}}(v)|)$ time since the set of vertices in $N_{\bar{H}}(v)$ which are labeled $nk + |C_k|$ is exactly the set $U \setminus N[C_k \cup \{v\}]$. If v is going to be added to C_k, then this increases the size of C_k by one and may affect the set $N[C_k]$. We update the labels of the vertices in $U \setminus N[C_k \cup \{v\}]$ by adding 1 to the label of every vertex in $N_{\bar{H}}(v)$ labeled with $nk + |C_k|$, and then we add v to C_k. This requires $O(|N_{\bar{H}}(v)|)$ time for each vertex and $O(|\bar{E}(H)| + |V(H)|)$ in total for the "defining P" part since every vertex is considered once and marked as a **p**-, a **c**-, or an **s**-vertex. The **s**-vertices may be reconsidered once and changed to **p**-vertices, but this does not affect the time complexity.

The "defining A" part consists of an if-else statement with 4 cases. In the first case we can do the required test by simply finding the largest neighborhood of a component and checking if its size is $|P|$. Without increasing the time complexity of the "defining P" part, we can store the values $|C|$ and $|U \setminus N_H[C]|$ for each component C of $H(U \setminus P)$. Thus $|N_H(C)| = |U| - (|C| + |U \setminus N_H[C]|)$.

In the second case, we check every nonedge in $H(P)$, which is also an $O(|\bar{E}(H)| + |V(H)|)$ operation.

In the third case we mark nonedges and components as follows: For each **p**-vertex u and then for each component C of $H(U \setminus P)$, where $C \subseteq N_{\bar{H}}(u)$, we mark C with the label u. We go through vertices in $N_{\bar{H}}(u)$, check which components they belong to, add up these numbers for each component, and check if it matches the total size of the component. Then for every **p**-vertex $v \in N_{\bar{H}}(u)$, where v is associated with a component labeled u, we add u to the label of nonedge uv. This takes $O(|N_{\bar{H}}(u)|)$ time for each **p**-vertex. The third case will now exist if and only if there is a nonedge uv marked by both u and v. Thus the total time for this case is $O(|\bar{E}(H)| + |V(H)|)$ for each subgraph H.

The fourth case requires constant time, and thus the total running time of Algorithm Partition on input subgraph H is $O(|\bar{E}(H)| + |V(H)|)$.

The operations that require $O(|\bar{E}(H)| + |V(H)|)$ on each subgraph H add up to $O(n^2)$ for all subgraphs of the same iteration of FMT since they do not overlap in nonedges, and there are at most $O(n)$ such graphs by Lemma 3.2. Thus the total time complexity for all subgraphs H at the same iteration is $O(n^2)$. □

We now give upper and lower bounds on the number of nonedges in various subgraphs of H related to vertex set P.

LEMMA 4.2. *Let P be as computed by Algorithm Partition(H). Then each of the following is true:*

(i) $|\bar{E}(H(N_H[C]))| \leq \frac{4}{5}|\bar{E}(H)|$ *for each connected component C of $H(U \setminus P)$.*

(ii) $|\bar{E}(H(N_H[v]))| > \frac{3}{5}|\bar{E}(H)|$ *for each* **s**-*vertex v.*

(iii) $|\bar{E}(H(N_H[C \cup \{v\}]))| > \frac{3}{5}|\bar{E}(H)|$ *for each* **p**-*vertex v associated with C, where C is a connected component of $H(U \setminus P)$.*

Proof. (i) From Algorithm Partition we know that $\mathcal{E}_{\bar{H}}(U \setminus N_H[C]) \geq \frac{2}{5}|\bar{E}(H)|$ for each connected component C of $H(U \setminus P)$. Each nonedge uv outside of $H(N_H[C])$ contributes to the degree-sum $\mathcal{E}_{\bar{H}}(U \setminus N_H[C])$ by 1 if one of either u or v is outside $N_H[C]$ and by 2 if both are outside. Thus there are at least $\frac{1}{5}|\bar{E}(H)|$ nonedges outside $H(N_H[C])$ and consequently at most $\frac{4}{5}|\bar{E}(H)|$ nonedges inside $H(N_H[C])$. Hence, $|\bar{E}(H(N_H[C]))| \leq \frac{4}{5}|\bar{E}(H)|$ for each connected component C of $H(U \setminus P)$, which completes the proof of (i).

(ii), (iii) From Algorithm Partition we know that $\mathcal{E}_{\bar{H}}(U \setminus N_H[v]) < \frac{2}{5}|\bar{E}(H)|$ for each **s**-vertex v and that $\mathcal{E}_{\bar{H}}(U \setminus N_H[C \cup \{u\}]) < \frac{2}{5}|\bar{E}(H)|$ for each **p**-vertex u associated with C. It follows by the same argument as case (i) that $|\bar{E}(H(N_H[v]))| > \frac{3}{5}|\bar{E}(H)|$ and $|\bar{E}(H(N_H[C \cup \{u\}]))| > \frac{3}{5}|\bar{E}(H)|$. This completes the proof of (ii) and (iii). □

We are now ready to prove the main result of this section, namely, that the vertex set A returned by Partition results in subproblems of size bounded by a constant factor of the number of nonedges, given in Theorem 4.3.

THEOREM 4.3. *Let A be the vertex set returned by Algorithm Partition on input $H = (U, D)$. Then both of the following are true where G' is as defined in Algorithm FMT:*

(i) *A is a proper subset of U such that either $A = N_H[K]$, where $K \subset U$ and $H(K)$ is connected, or A is a pmc of H.*

(ii) *Both the number of nonedges in $G'(A)$ and the number of nonedges in $G'(N_H[C])$ for each connected component C of $H(U \setminus A)$ are at most $\frac{4}{5}|\bar{E}(H)|$.*

Proof. We will examine each of the 4 cases of the if-else statement in the "defining A" part of Algorithm Partition. We omit the subscript H in $N_H(C)$ and $N_H[C]$ to increase readability. The reader should keep in mind that throughout this proof we regard neighborhoods in H (and not in $\bar{H}$).

Case 1. $H(U \setminus P)$ has a full component C, i.e., $P = N(C)$.

This implies in particular that no vertices could have been marked as **s**-vertices. By Lemma 4.2 we know that the number of nonedges in $H(N[C_i])$ is less than $\frac{4}{5}|\bar{E}(H)|$ for each connected component C_i of $H(U \setminus P)$, in particular, for C. In this case, Algorithm Partition gives $A = N[C]$, and thus $P \subset A$. C is a connected set since it was computed by adding new members from its neighborhood, and so (i) is satisfied. Observe that the connected components C_i of $H(U \setminus A)$ are exactly the connected components C_i of $H(U \setminus P)$, except C. It follows that the number of nonedges in $H(A) = H(N[C_i])$ and in $H(N[C_j])$ for each connected component C_j of $H(U \setminus A)$ is less than $\frac{4}{5}|\bar{E}(H)|$ already before the minimal separators are saturated. After the saturation, this number cannot increase but only decrease.

Case 2. There exist two vertices u, v such that $uv \notin E(H)$, u is marked as an **s**-vertex, and v is marked as an **s**-vertex or a **p**-vertex.

We give the proof in two parts: the subcase where both u, v are **s**-vertices and the subcase where u is an **s**-vertex and v a **p**-vertex. The arguments for the two subcases are very similar, and note that they are also very similar to Case 3, where both u, v are **p**-vertices.

Assume both u and v are marked as **s**-vertices. By Lemma 4.2, $|\bar{E}(H(N[u]))| > \frac{3}{5}|\bar{E}(H)|$ and $|\bar{E}(H(N[v]))| > \frac{3}{5}|\bar{E}(H)|$, and thus for their common part we have $|\bar{E}(H(N(u) \cap N(v)))| = |\bar{E}(H(N[u] \cap N[v]))| > \frac{1}{5}|\bar{E}(H)|$, where the first equality holds since $u \notin N[v]$. The algorithm gives $A = N[u]$ in this case, satisfying (i), which means that v will belong to a component C of $H(U \setminus A)$ with $N(C) \subseteq A$ thus being a u, v-separator. Since any u, v-separator must contain $N(u) \cap N(v)$, it follows that $N(C) \subseteq A$ induces at least $\frac{1}{5}|\bar{E}(H)|$ nonedges. All these nonedges will become edges and disappear from G'. Thus, there are at most $\frac{4}{5}|\bar{E}(H)|$ nonedges left that can appear in subproblems $G'(A)$ or $H(N[C_i])$ for a component C_i of $H(U \setminus A)$, thereby also satisfying (ii).

Assume u is marked as an **s**-vertex and v is marked as a **p**-vertex, and let j be the index such that v is associated to C_j. We know that such a C_j exists since v is marked as a **p**-vertex. An important observation now is that $u \notin N[C_j \cup \{v\}]$. Otherwise u would have been marked as a **p**-vertex or **c**-vertex during execution of the inner while-loop in Algorithm Partition during computation of C_j. By Lemma 4.2, $|\bar{E}(H(N[u]))| > \frac{3}{5}|\bar{E}(H)|$ and $|\bar{E}(H(N[C_j \cup \{v\}]))| > \frac{3}{5}|\bar{E}(H)|$, and thus for their common part we have $|\bar{E}(H(N(u) \cap N(C_j \cup \{v\})))| = |\bar{E}(H(N[u] \cap N[C_j \cup \{v\}]))| > \frac{1}{5}|\bar{E}(H)|$, where the first equality holds since $u \notin N[C_j \cup \{v\}]$, as we established above. The algorithm gives $A = N[u]$ in this case, satisfying (i), which means that $C_j \cup \{v\}$ will be contained in a component C of $H(U \setminus A)$ with $N(C) \subseteq A$, thus separating u from $C_j \cup \{v\}$. Since any such separator must contain $N(u) \cap N(C_j \cup \{v\})$, it follows that $N(C) \subseteq A$ induces at least $\frac{1}{5}|\bar{E}(H)|$ nonedges. All these nonedges will become edges and disappear from G'. Thus, there are at most $\frac{4}{5}|\bar{E}(H)|$ nonedges left that can appear in subproblems $G'(A)$ or $H(N[C_i])$ for a component C_i of $H(U \setminus A)$, thereby also satisfying (ii).

Case 3. There exist two vertices u, v marked as **p**-vertices such that $uv \notin E(H)$, u is associated with C_i, v is associated with C_j, $u \notin N(C_j)$, and $v \notin N(C_i)$.

The important observation now is that there are no edges between $C_i \cup \{u\}$ and $C_j \cup \{v\}$. By Lemma 4.2, $|\bar{E}(H(N[C_i \cup \{u\}]))| > \frac{3}{5}|\bar{E}(H)|$ and $|\bar{E}(H(N[C_j \cup \{v\}]))| > \frac{3}{5}|\bar{E}(H)|$, and thus for their common part we have $|\bar{E}(H(N(C_i \cup \{u\}) \cap N(C_j \cup \{v\})))| = |\bar{E}(H(N[C_i \cup \{u\}] \cap N[C_j \cup \{v\}]))| > \frac{1}{5}|\bar{E}(H)|$, where the first equality holds since there are no edges between $C_i \cup \{u\}$ and $C_j \cup \{v\}$. The algorithm gives $A = N[C_i \cup \{u\}]$ in this case, satisfying (i), which means that $C_j \cup \{v\}$ will be contained in a component C of $H(U \setminus A)$ with $N(C) \subseteq A$, thus separating $C_i \cup \{u\}$ from $C_j \cup \{v\}$. Since any such separator must contain $N(C_i \cup \{u\}) \cap N(C_j \cup \{v\})$, it follows that $N(C) \subseteq A$ induces at least $\frac{1}{5}|\bar{E}(H)|$ nonedges. All these nonedges will become edges and disappear from G'. Thus, there are at most $\frac{4}{5}|\bar{E}(H)|$ nonedges left that can appear in subproblems $G'(A)$ or $H(N[C_i])$ for a component C_i of $H(U \setminus A)$, thereby also satisfying (ii).

Case 4. None of the above cases apply.

First we show that P is a pmc of H in this case. Due to Theorem 2.1, all we have to show is that if none of the Cases 1, 2, and 3 apply, then $H(U \setminus P)$ has no full component associated with P, and for every pair of nonadjacent vertices $u, v \in P$, there is a connected component C of $H(U \setminus P)$ such that $u, v \in N(C)$. Since Case 1 does not apply, we know that $H(U \setminus P)$ has no full components. Since Case 2 does not apply either, then the **s**-vertices altogether induce a clique, and they all have edges to all **p**-vertices. So, since P consists only of **p**- and **s**-vertices, the only nonedges that are possible within P are those nonedges uv where both u and v are **p**-vertices. Since Case 3 does not apply either, then for any nonadjacent $u, v \in P$, if they are not associated with the same component, then one of them must be in the neighborhood of the component that the other one is associated with. Thus P is a pmc of H, and (i) is satisfied since Algorithm Partition gives $A = P$ in this case. In this case, whole A is saturated in G', and thus $G'(A)$ has no nonedges. The remaining subproblems will each have at most $\frac{4}{5}|\bar{E}(H)|$ nonedges by Lemma 4.2 since the connected components of $H(U \setminus A)$ are the same as the connected components of $H(U \setminus P)$. □

5. The total $O(n^\alpha \log n)$ time complexity.

THEOREM 5.1. *Algorithm FMT described in section* 3*, using Algorithm Partition described in section* 4*, computes a minimal triangulation of the input graph in* $O(n^\alpha \log n)$ *time.*

Proof. By Lemma 3.1 and Theorem 4.3(i), Algorithm FMT computes a minimal triangulation. By Lemma 3.2, the matrix multiplication at each iteration of FMT requires $O(n^\alpha)$ time. By the discussion that follows Lemma 3.2 in section 3, all other operations outside of Algorithm Partition can be performed in $O(n^2)$ time at each iteration of FMT. Using Lemma 4.1, we conclude that the total time required at each iteration of FMT is $O(n^\alpha)$ since $\alpha \geq 2$ for any matrix multiplication algorithm. By Theorem 4.3(ii), the number of nonedges in each subproblem decreases by a constant factor for each iteration, and since subproblems in one iteration do not overlap in nonedges, we can have at most $\log n^2 = O(\log n)$ iterations of FMT. □

We have thus given the details of a new algorithm to compute minimal triangulations of arbitrary graphs in $O(n^\alpha \log n)$ time. It is important to use a matrix multiplication algorithm with running time $o(n^3)$ to achieve an improvement compared to existing minimal triangulation algorithms, and thus standard matrix multiplication is not interesting. If we use the matrix multiplication algorithm of Coppersmith and Winograd [6], then α is strictly less than 2.376, and thus the total running time of our algorithm becomes $o(n^{2.376})$. If we instead use the matrix multiplication algorithm of Strassen [18], which has a worse asymptotic time bound of $\Theta(n^{\log_2 7}) = o(n^{2.81})$ but is considered more practical due to large constants in [6], then our time bound becomes

$O(n^{\log_2 7} \log n) = o(n^{2.81})$. Using Strassen's algorithm, the time bound claimed by Kratsch and Spinrad [12] mentioned previously becomes $O(n^{2.91})$. In fact, our algorithm is asymptotically faster than theirs regardless of the matrix multiplication algorithm used.

REFERENCES

[1] A. Berry, J.-P. Bordat, and P. Heggernes, *Recognizing weakly triangulated graphs by edge separability*, Nordic J. Comput., 7 (2000), pp. 164–177.

[2] A. Berry, J.-P. Bordat, P. Heggernes, G. Simonet, and Y. Villanger, *A wide-range algorithm for minimal triangulation from an arbitrary ordering*, J. Algorithms, to appear.

[3] A. Berry, P. Heggernes, and Y. Villanger, *A vertex incremental approach for dynamically maintaining chordal graphs*, in Algorithms and Computation—ISAAC 2003, Lecture Notes Comput. Sci. 2906, Springer-Verlag, Berlin, 2003, pp. 47–57.

[4] J. R. S. Blair, P. Heggernes, and J. A. Telle, *A practical algorithm for making filled graphs minimal*, Theoret. Comput. Sci., 250 (2001), pp. 125–141.

[5] V. Bouchitté and I. Todinca, *Treewidth and minimum fill-in: Grouping the minimal separators*, SIAM J. Comput., 31 (2001), pp. 212–232.

[6] D. Coppersmith and S. Winograd, *Matrix multiplication via arithmetic progressions*, J. Symbolic Comput., 9 (1990), pp. 1–6.

[7] E. Dahlhaus, *Minimal elimination ordering inside a given chordal graph*, in Graph-Theoretic Concepts in Computer Science—WG '97, Lecture Notes in Comput. Sci. 1335, Springer-Verlag, Berlin, 1997, pp. 132–143.

[8] P. Heggernes and Y. Villanger, *Efficient implementation of a minimal triangulation algorithm*, in Algorithms—ESA 2002, Lecture Notes in Comput. Sci. 2461, Springer-Verlag, Berlin, 2002, pp. 550–561.

[9] D. Hudson, S. Nettles, and T. Warnow, *Obtaining highly accurate topology estimates of evolutionary trees from very short sequences*, in Proceedings of RECOMB'99, 1999, pp. 198–207.

[10] T. Kloks, D. Kratsch, and J. Spinrad, *On treewidth and minimum fill-in of asteroidal triple-free graphs*, Theoret. Comput. Sci., 175 (1997), pp. 309–335.

[11] D. Kratsch and J. Spinrad, *Between $O(nm)$ and $O(n^{\alpha})$*, in Proceedings of the 14th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA 2003), SIAM, Philadelphia, 2003, pp. 709–716.

[12] D. Kratsch and J. Spinrad, *Minimal fill in $o(n^3)$ time*, Discrete Math., submitted.

[13] T. Ohtsuki, *A fast algorithm for finding an optimal ordering for vertex elimination on a graph*, SIAM J. Comput., 5 (1976), pp. 133–145.

[14] T. Ohtsuki, L. K. Cheung, and T. Fujisawa, *Minimal triangulation of a graph and optimal pivoting ordering in a sparse matrix*, J. Math. Anal. Appl., 54 (1976), pp. 622–633.

[15] A. Parra and P. Scheffler, *Characterizations and algorithmic applications of chordal graph embeddings*, Discrete Appl. Math., 79 (1997), pp. 171–188.

[16] B. W. Peyton, *Minimal orderings revisited*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 271–294.

[17] D. Rose, R. E. Tarjan, and G. Lueker, *Algorithmic aspects of vertex elimination on graphs*, SIAM J. Comput., 5 (1976), pp. 266–283.

[18] V. Strassen, *Gaussian elimination is not optimal*, Numer. Math., 14 (1969), pp. 354–356.

[19] J. van Leeuwen, *Graph algorithms*, in Handbook of Theoretical Computer Science, A: Algorithms and Complexity Theory, North–Holland, Amsterdam, 1990.

SIAM J. DISCRETE MATH.
Vol. 19, No. 4, pp. 914–920

SHORT ANSWERS TO EXPONENTIALLY LONG QUESTIONS: EXTREMAL ASPECTS OF HOMOMORPHISM DUALITY[*]

JAROSLAV NEŠETŘIL[†] AND CLAUDE TARDIF[‡]

Abstract. We prove that there exists a constant k such that for every $n \geq 1$ there exists a directed core graph H_n with at least 2^n vertices such that a directed graph G is H_n-colorable if and only if every subgraph of G with at most $kn\log(n)$ vertices is H_n-colorable. Our examples show that in general the "duals of relational structures" in the sense of [J. Nešetřil and C. Tardif, *J. Combin. Theory Ser. B*, 80 (2000), pp. 80–97] can have superpolynomial size. The construction given in this paper gives a double exponential upper bound for such a construction. Here we improve this to an exponential upper bound.

Key words. homomorphism duality, graphs, relational structures, duals, colorings, finite models

AMS subject classifications. Primary, 05C15; Secondary, 68R05, 03C13, 06A07, 05E99

DOI. 10.1137/S0895480104445630

1. Introduction.

A *homomorphism* between two directed graphs G and H is a map ϕ from the vertex set of G to that of H such that $(\phi(x), \phi(y))$ is an arc of H whenever (x, y) is an arc of G. We write $G \to H$ when there exists a homomorphism from G to H. For a fixed target H, the *H-coloring problem* is the following decision problem:

***H*-coloring problem**
Instance: A directed graph G.
Question: Does there exist a homomorphism from G to H?

The complexity of the H-coloring problem depends on H. A complete classification seems out of reach for the moment, but the dichotomy conjecture of [2] (see also [1]) states that every H-coloring problem is polynomial or NP-complete.

Here we concentrate on a subclass of the polynomial H-coloring problems, namely, those for which there exists a constant $m(H)$ such that the following holds.

For every directed graph G, there exists a homomorphism from G to H if and only if every subgraph G' of G with at most $m(H)$ vertices admits a homomorphism to H.

The H-coloring problem can then be reduced to a polynomial search for an obstruction to a homomorphism among the subgraphs of G with at most $m(H)$ vertices. The best known example of this situation is the relation between the transitive tournament on n vertices and the directed path with n forward edges (see [3, 4, 14, 16]): A directed graph G admits a homomorphism to the former if and only if it admits no homomorphism from the latter; hence it is sufficient to look for an obstruction among the subgraphs of G with at most $n + 1$ vertices.

[*] Received by the editors August 10, 2004; accepted for publication (in revised form) May 4, 2005; published electronically December 30, 2005.
http://www.siam.org/journals/sidma/19-4/44563.html

[†] Department of Applied Mathematics and Institute for Theoretical Computer Science (ITI), Charles University, Malostranské nám.25, 11800 Praha 1, Czech Republic (nesetril@kam.ms.mff.cuni.cz). The work of this author was partially supported by project 1M0021620808 of the Czech Ministry of Education.

[‡] Department of Mathematics and Computer Science, Royal Military College of Canada, Kingston, ON K7K 7B4, Canada (Claude.Tardif@rmc.ca). The work of this author was partially supported by grant 261525-03 of the Natural Sciences and Engineering Research Council of Canada and by a grant from the Academic Research Program.

More generally, for any H-coloring problem considered here, there is only a finite list $O_1, O_2, \ldots, O_m$ of directed graphs with at most $m(H)$ vertices which do not admit a homomorphism to H. According to [11, Theorems 2.9, 3.1], the "minimal" obstructions among these are directed trees $T_1, \ldots, T_\ell$. For each tree T_i, there exists a "dual" directed graph D_i with the following property:

For every directed graph G, there exists a homomorphism from G to
D_i if and only if there exists no homomorphism from T_i to G.

H is then homomorphically equivalent to the product of these duals.

The construction given in [11] for the dual of a tree T gives a directed graph D which could have as many as $2^{2^{|V(T)|}}$ vertices, yielding $m(D) \simeq \lg(\lg(|V(D)|))$. However, in the example cited above, where T is the directed path with n forward arcs and D is the transitive tournament with n vertices, we have $m(D) = |V(D)| + 1$. Indeed in all known cases, the dual D of T can be "dismantled" to a structure with the same order of magnitude as T. Thus questions arise as to whether polynomial constructions would be possible instead of the double exponential construction of [11].

In this paper, we answer these questions by giving a new construction which always gives a dual with at most $2^{n \lg(n)}$ vertices for a tree with n vertices. The new construction is conceptually simpler and yields new insights in the structure of duals (see [15]). On the other hand, we can also exhibit trees with n vertices whose dual must have at least $2^{\Omega(n/\lg(n))}$ vertices, indicating that the new construction is close to optimal.

Our construction will be presented in the general context of relational structures, that is, the original context of [11], which is also the natural context of constraint satisfaction problems [1, 2]. Incidentally, we note that it is a specification of relational examples that led to the discovery of the examples mentioned above. We give the necessary terminology in the following section. The new construction of duals is given in section 3, and the examples with large duals are given in section 4. We will conclude with a few comments concerning the bound $m(H)$.

2. Relational structures. Let $\Delta = (\delta_i; i \in I)$ be a sequence of positive integers. A *relational structure of type* Δ (or Δ-structure) is a pair $A = (X, (R_i; i \in I))$ where X is a finite set and R_i is a δ_i-nary relation on X (that is, $R_i \subset X^{\delta_i}$). We will denote $\underline{A}$ the base set of A (that is, $\underline{A} = X$ when $A = (X, (R_i; i \in I))$).

Given a type Δ and Δ-structures $A = (X, (R_i; i \in I))$ and $A' = (X', (R'_i; i \in I))$, a *homomorphism* from A to A' is a mapping $f : X \mapsto X'$ such that for every $i \in I$ we have

$$(f(x_1), f(x_2), \ldots, f(x_{\delta_i})) \in R'_i \quad \text{whenever } (x_1, x_2, \ldots, x_{\delta_i}) \in R_i.$$

We write $A \to A'$ if there exists a homomorphism from A to A'.

For a Δ-structure H, the H-coloring problem is defined just as in the case of directed graphs:

***H*-coloring problem**
Instance: A Δ-structure A.
Question: Does there exist a homomorphism from A to H?

Two Δ-structures H and H' are called *homomorphically equivalent* if $H \to H'$ and $H' \to H$; we then write $H \leftrightarrow H'$. Note that when $H \leftrightarrow H'$, we have $A \to H$ if and only if $A \to H'$; hence the H-coloring problem is equivalent to the H'-coloring problem.

A Δ-structure is called a *core* if it is not homomorphically equivalent to any Δ-structure on a smaller base set. Clearly, any Δ-structure is homomorphically equivalent to at least one core. It can be shown (see [11]) that two homomorphically equivalent cores are isomorphic; hence the core of any Δ-structure H is well defined up to isomorphism. In studying H-coloring problems, we can restrict our attention to the case where H is a core without loss of generality. Indeed, the parameter $m(H)$ presented in the introduction is not very interesting when H is not a core.

3. A construction of duals. Let A be a relational structure of type $\Delta = (\delta_i; i \in I)$. We define the *incidence graph* $\mathrm{Inc}(A)$ of A as the bipartite graph with parts $\underline{A}$ and

$$\mathrm{Block}(A) = \{(i, (a_1, \dots, a_{\delta_i})) : i \in I, (a_1, \dots, a_{\delta_i}) \in R_i(A)\}$$

and edges $[a, (i, (a_1, \dots, a_{\delta_i}))]$ for every $k \in \{1, \dots, \delta_i\}$ such that $a = a_k$. (Hence $\mathrm{Inc}(A)$ could be a multigraph.) A is called a Δ-*tree* when $\mathrm{Inc}(A)$ is a tree (hence it contains no 2-cycles, that is, multiple edges).

A Δ-structure D is called a *dual* of A if for every Δ-structure X, there exists a homomorphism $\phi : X \mapsto D$ if and only if there is no homomorphism $\phi : A \mapsto X$. In [11], it was shown that a structure A admits a dual if and only if A is a Δ-tree.[1] Note that any two duals D, D' of A are necessarily homomorphically equivalent. Therefore it is possible to define *the* dual of a Δ-tree A up to homomorphic equivalence. In [11] a construction for duals of Δ-trees is presented, using gaps and exponentiation. In some cases this construction will yield a structure of size in the order of $2^{2^{|\underline{A}|+|\Delta|}}$ as the dual of a Δ-tree A. We present here a new construction that is conceptually simpler and always yields duals of size at most $2^{(|\underline{A}|+|\Delta|)\lg|\underline{A}|}$.

Definition 1. *Let A be a Δ-tree. We define $D(A)$ as the structure defined on the base set*

$$\underline{D(A)} = \{f : \underline{A} \mapsto \mathrm{Block}(A) : \ [a, f(a)] \in E(\mathrm{Inc}(A)) \textit{ for all } a \in \underline{A}\}$$

by putting $(f_1, \dots, f_{\delta_i})$ in $R_i(D(A))$ if and only if for all $(x_1, \dots, x_{\delta_i}) \in R_i(A)$ there exists $j \in \{1, \dots, \delta_i\}$ such that $f_j(x_j) \neq (i, (x_1, \dots, x_{\delta_i}))$.

Note that $\underline{D(A)}$ has at most $|\underline{A}|^{|\underline{A}|+|\Delta|}$ elements. We prove that $D(A)$ is indeed a dual of A.

Theorem 2. *Let A be a Δ-tree. Then for every Δ-structure X, there exists a homomorphism from X to $D(A)$ if and only if there is no homomorphism from A to X.*

Proof. We first prove by contradiction that there is no homomorphism from A to $D(A)$. Suppose that there exists a homomorphism $\phi : A \mapsto D(A)$; for all $a \in \underline{A}$, put $f_a = \phi(a)$. We fix $a_0 \in \underline{A}$ and define a sequence $(a_k)_{k \geq 0}$ recursively as follows: If $f_{a_k}(a_k) = (i, (x_1, \dots, x_{\delta_i}))$, then since ϕ is a homomorphism, we have $(f_{x_1}, \dots, f_{x_{\delta_i}}) \in R_i(D(A))$; hence $f_{x_j}(x_j) \neq (i, (x_1, \dots, x_{\delta_i}))$ for some $j \in \{1, \dots, \delta_i\}$. We then put $a_{k+1} = x_j$. The sequence $a_0, f_{a_0}(a_0), a_1, f_{a_1}(a_1), a_2, \dots$ is then a trail in $\mathrm{Inc}(A)$ such that $a_{k+1} \neq a_k$ and $f_{a_{k+1}}(a_{k+1}) \neq f_{a_k}(a_k)$ for all $k \geq 0$, which is impossible since $\mathrm{Inc}(A)$ is a finite tree. Therefore there is no homomorphism from A to $D(A)$; consequently if a Δ-structure X admits a homomorphism from A, then there is no homomorphism from X to $D(A)$. This concludes the first part of the proof.

For the second part of the proof, we will need to fix some notation. For a in $\underline{A}$ and a neighbor $b = (i, (x_1, \dots, x_{\delta_i}))$ of a in $\mathrm{Inc}(A)$, let $T_{a,b}$ be the maximal subtree of

[1] The definition of Δ-trees given in [11] is a bit different from the one given here, but it is not hard to show that the two definitions are equivalent.

$\mathrm{Inc}(A)$ containing a and b but no other neighbor of a, and let $A_{a,b}$ be the Δ-subtree of A such that $\mathrm{Inc}(A_{a,b}) = T_{a,b}$. Thus, for a fixed a we have $A = \cup\{A_{a,b} : b \in N_{T_{a,b}}(a)\}$, where for $b \neq b'$ we have $A_{a,b} \cap A_{a,b'} = \{a\}$.

We also fix a vertex-labeling $\ell : \mathrm{Inc}(A) \mapsto \mathbb{N}$ with the following properties:

- $u \neq v$ implies $\ell(u) \neq \ell(v)$;
- for all $n \in \mathbb{N}$, $\{u : \ell(u) \geq n\}$ induces a connected subtree of $\mathrm{Inc}(A)$.

(Such an ℓ is easily defined by repeatedly labeling and deleting the pendant vertices of $\mathrm{Inc}(A)$.)

Now, let X be a Δ-structure such that there is no homomorphism from A to X. For every $x \in \underline{X}$ and $a \in \underline{A}$, there necessarily exists a b adjacent to a in $\mathrm{Inc}(A)$ such that there is no homomorphism ψ_b from $A_{a,b}$ to X with $\psi(a) = x$ (for otherwise the union of all of these ψ_b would be a homomorphism from A to X). We fix $f_x(a)$ to be such a b with the smallest label. This allows us to define a function $\phi : \underline{X} \mapsto \underline{D(A)}$ by $\phi(x) = f_x$; we will show that it is a homomorphism from X to $D(A)$.

We need to show that for $i \in I$ and $(x_1, \dots, x_{\delta_i}) \in R_i(X)$, we have $(f_{x_1}, \dots, f_{x_{\delta_i}}) \in R_i(D(A))$. By definition of $D(A)$, we have $(f_{x_1}, \dots, f_{x_{\delta_i}}) \in R_i(D(A))$ if and only if for every $(a_1, \dots, a_{\delta_i}) \in R_i(A)$, there exists an index j such that $f_{x_j}(a_j) \neq (i, (a_1, \dots, a_{\delta_i}))$. (It is worthwhile to note that at this point in the proof, a medium-sized brown bear burst into the office and made its way to the coffee table in the corner. Though not particularly ferocious, this animal can be irritated by the presence of a human, and the authors were left with no other recourse than to climb atop filing cabinets and wait until the proper authorities came in and restored the beast to its natural habitat. Overall, the incident can only be described as disquieting.) We proceed to prove that ϕ is a homomorphism by contradiction, assuming that for some $(a_1, \dots, a_{\delta_i}) \in R_i(A)$, we have $f_{x_j}(a_j) = b = (i, (a_1, \dots, a_{\delta_i}))$ for all $j \in \{1, \dots, \delta_i\}$. Note that there exists at most one index j such that a_j is adjacent to some b' such that $\ell(b') > \ell(b)$. For every other index k and every $b' \neq b$ adjacent to a_k, we have $\ell(b') < \ell(b)$; therefore there exists a homomorphism $\psi_{a_k,b'} : A_{a_k,b'} \mapsto X$ such that $\psi_{a_k,b'}(a_k) = x_k$. The union of all these $\psi_{a_k,b'}$ is a well-defined map ψ from some subset of $\underline{A}$ to $\underline{X}$. Now if no index j fits the description given above, then ψ is in fact a homomorphism from A to X, which contradicts the fact that no such homomorphism exists. On the other hand, if some index j fits this description, then putting $\psi(a_j) = x_j$ turns ψ into a homomorphism from $A_{a_j,b}$ to X such that $\psi(a_j) = x_j$, contradicting the definition of $f_{x_j}(a_j)$. Therefore the δ_i-tuple $(a_1, \dots, a_{\delta_i})$ described above cannot exist; hence ϕ is a homomorphism from X to $D(A)$. ◻

4. Paths with large duals. In [11] we constructed trees with exponentially large dual cores. More precisely, we constructed a Δ-tree T of type $\Delta = (1, 1, \dots, 1, n)$ with n unary relations such that T has n vertices and its dual D_T has a core with 2^n vertices. The possible existence of large dual cores for a *fixed* type Δ was left as an open problem. Here we answer this question positively, even for the simplest type (2) corresponding to directed graphs.

We proceed in two steps: First we consider the type $\Delta_n = (2, 2, \dots, 2)$ (i.e., relational systems with $n + 1$ binary relations) and then we modify this to the type (2).

DEFINITION 3. *Let $n > 2$ be an integer. We define P_n as the structure of the type Δ_n with $n+1$ binary relations $R_0, R_1, \dots, R_n$ on the base set $\underline{T_n} = \{x_0, y_0, x_1, y_1, \dots, x_n, y_n\}$ given by*

(i) $R_0(P_n) = \{(x_i, y_i) : i = 0, \dots, n\}$,

(ii) $R_i(P_n) = \{(y_{i-1}, x_i)\}, i = 1, \dots, n.$

In what follows, D_n will denote the core of the dual of P_n. We will prove the following.

LEMMA 4. *For every $S \subseteq \{1, 2, \ldots, n\}$, there exists an element $f_S \in \underline{D_n}$ such that $(f_S, f_S) \in R_i(D_n)$ if $i \in S$ and $(f_S, f_S) \notin R_i(D_n)$ if $i \in \{1, 2, \ldots, n\} \setminus S$.*

Proof. For $i \in \{1, 2, \ldots, n\}$, let Q_i be the structure obtained from P_n by removing the arc (y_{i-1}, x_i) from R_i and identifying y_{i-1}, x_i in a new point labeled t. Now for $S \subseteq \{1, 2, \ldots, n\}$, let L_S be the structure obtained from the disjoint union of all $Q_i : i \notin S$ by identifying all points labeled t and adding the loop (t, t) in $R_i(L_S)$ for all $i \in S$. By construction, we then have $P_n \not\to L_S$, but adding the loop (t, t) in $R_i(L_S)$ for any $i \notin S$ would produce a structure admitting a homomorphism from P_n. Therefore, there exists a homomorphism $\phi : L_S \mapsto D_n$, and $f = \phi(t)$ satisfies $(f, f) \in R_i(D_n)$ for all $i \in S$, and $(f, f) \notin R_i(D_n)$ for all $i \in \{1, 2, \ldots, n\} \setminus S$. □

Thus for distinct subsets $S, S' \subseteq \{1, 2, \ldots, n\}$, we have $f_S \neq f_{S'}$, which implies the following corollary.

COROLLARY 5. $|\underline{D_n}| \geq 2^n$.

In Corollary 5, we use $n + 1$ binary relations to construct a path whose dual has 2^n elements, which leaves open the possibility that polynomial constructions exist for every fixed type. In the remainder of this section, we will modify this construction to build directed graphs with superpolynomial duals.

LEMMA 6. *Let $n > 2$ be a fixed integer. Then there exist paths $MR_0, MR_1, \ldots, MR_n$, each with $3\lceil \lg(n + 1)\rceil + 4$ arcs, such that there exists a homomorphism from MR_i to MR_j if and only if $i = j$.*

Proof. For simplicity suppose that $n + 1 = 2^m$. Let A_0 be the path consisting of one backward edge followed by two forward edges, let A_1 be the path consisting of two forward edges followed by one backward edge, and let A_2 be the path consisting of two forward edges. Then, every $i \in \{0, \ldots, n\}$ corresponds to a sequence $(\epsilon_1, \ldots, \epsilon_m) \in \{0, 1\}^m$. We then define

$$MR_i = A_2 \circ A_{\epsilon_1} \circ A_{\epsilon_2} \circ \cdots \circ A_{\epsilon_m} \circ A_2,$$

where the concatenation $A_x \circ A_y$ is obtained simply by identifying the last vertex of A_x to the first vertex of A_y. Any homomorphism ϕ from MR_i to MR_j must preserve the algebraic length (that is, the difference between the number of forward edges and the number of backward edges) on any path; hence ϕ must map the initial vertex of MR_i to the initial vertex of MR_j and the terminal vertex of MR_i to the terminal vertex of MR_j. Therefore ϕ must be bijective and hence an isomorphism, which implies that $i = j$. □

The notation MR_i stands for "mock R_i." Given a structure X of type Δ_n with $n + 1$ binary relations, we will construct a directed graph $G(X)$ which encodes the structure of X as follows: For each u in $\underline{X}$, $G(X)$ contains a path MV_u starting at a vertex labeled IN followed by one backward arc, six forward arcs, and one backward arc, terminating at a vertex labeled OUT. For each $(u, v) \in R_i(X)$, we add a copy of MR_i to $G(X)$, identifying its initial vertex to the OUT vertex of MV_u, and its terminal vertex to the IN vertex of MV_v. This construction has the following property.

LEMMA 7. *Let X, Y be structures of type Δ_n (where $n > 2$). Then there exists a homomorphism from X to Y if and only if there exists a homomorphism from $G(X)$ to $G(Y)$.*

Proof. By construction, a homomorphism $\phi : X \mapsto Y$ naturally induces a homomorphism $\psi : G(X) \mapsto G(Y)$. Conversely, suppose that there exists a homomorphism $\psi : G(X) \mapsto G(Y)$. The directed paths of length 6 in $G(X)$ are the paths consisting of inner arcs in the subgraphs $MV_u : u \in \underline{X}$, and these must be mapped by ψ to directed paths of length 6 in $G(Y)$ which are precisely the paths consisting of inner arcs in the subgraphs $MV_v : v \in \underline{Y}$. Therefore we can define a map $\phi : \underline{X} \mapsto \underline{Y}$ by putting $\phi(u) = v$ if $\psi(MV_u) \subseteq MV_v$. Lemma 6 and the construction of $G(X)$ and $G(Y)$ then imply that ϕ is a homomorphism. □

Note that for the structure P_n of Definition 3, $G(P_n)$ is a path with $8 \cdot (2n+2) + (3\lceil \lg(n+1) \rceil + 4) \cdot (2n+1) = \Theta(n \lg(n))$ arcs. Let D'_n be the core of $D(G(P_n))$; we will prove the following theorem.

THEOREM 8. $|\underline{D'_n}| \geq 2^n$.

Proof. For each structure $L_S, S \subseteq \{1, \dots, n\}$ defined in the proof of Lemma 4, we have $G(P_n) \not\to G(L_S)$; thus there exists a homomorphism $\phi_S : G(L_S) \mapsto D'_n$. The distinguished element t of L_S corresponds to the path MV_t in $G(L_S)$; we denote m_S the midpoint of this path. For $S \neq S'$ we must have $\phi_S(m_S) \neq \phi_{S'}(M_{S'})$ for otherwise the combined cycles would imply the existence of a homomorphism from $G(P_n)$ to D'_n, just as in the proof of Lemma 4. Therefore $|\underline{D'_n}| \geq |\mathcal{P}(\{1, \dots, n\})| = 2^n$. □

Note that if k denotes the number of vertices in $G(P_n)$, then D'_n must have order $2^{\Omega(k/\lg(k))}$ as claimed.

5. Concluding comments. For a directed graph H, the parameter $m(H)$ discussed in the introduction can be defined as the "maximal size of an H-critical graph":

$$m(H) = \max\{|V(G)| : \ G \not\to H \text{ and } G' \to H \text{ for every proper subgraph } G' \text{ of } G\}.$$

We define the function $m^* : \mathbb{N} \mapsto \mathbb{N}$ by

$$m^*(n) = \min\{m(H) : \ H \text{ is a core and } |V(H)| = n\}.$$

The example of transitive tournaments shows that $m^*(n) \leq n + 1$, and the graphs D'_n of Theorem 8 lower this bound to $m^*(n) \in O(\lg(n) \lg(\lg(n)))$. In a sense this is counterintuitive, since it proves that there are directed graphs H for which the H-coloring problem is decided by obstructions much smaller than H. However, the true order of m^* may be smaller still.

The *categorical (direct) product* $\Pi_{i=1}^{\ell} H_i$ of a family $\{H_i\}_{i \in \{1,\dots,\ell\}}$ of directed graphs is the directed graph whose vertices are the n-tuples $u \in \Pi_{i=1}^{\ell} V(H_i)$, and whose arcs are the couples (u, v) such that (u_i, v_i) is an arc of H_i for all i in $\{1, \dots, \ell\}$. Let H be a directed core for which $m(H)$ is finite. By [11, Theorems 2.9, 3.1], there exist trees $T_1, \dots, T_\ell$ such that $H \leftrightarrow \Pi_{i=1}^{\ell} D(T_i)$. Putting $n = |V(H)|$ and $m = \max\{|V(T_i)| : 1 \leq i \leq \ell\}$, we have $\ell \leq 2^{m-1} m^{m-2}$ by Cayley's tree enumeration formula, whence $n \leq 2^{(2m)^{m-1} \lg(m)}$ by Theorem 2. This shows that $m^*(n) \in \Omega(\lg(\lg(n))/ \lg(\lg(\lg(n))))$.

At the moment it is not known which of the logarithmic upper bound and the double logarithmic lower bound is closer to the true order of m^*. The question depends on finding bounds on cores H_m of products $\Pi\{D(T) : T \in \mathcal{F}_m\}$, where $\mathcal{F}_m$ is an exponential family of m-trees. On one hand, finding infinite families of examples where $|V(H_m)| \in \Omega\left(2^{2^m}\right)$ would prove a double logarithmic behavior for m^*. On the other hand, if such families are hard to find, then there may be many infinite families of examples where $|V(H_m)| \in O(2^m)$. Now consider the following decision problem:

Instance: A directed graph G and an integer m.

Question: Does there exist a homomorphism from G to H_m?

The problem is in Co-NP since a homomorphism from a member of $\mathcal{F}_m$ is a polynomial certificate for a negative answer. If $|V(H_m)| \in O(2^m)$ and a polynomial description of vertices and adjacencies in H_m exists, then the problem is also in NP. Hence the hypothesis that $m^*(n) \in \Omega(\lg(n))$ would suggest that many such intriguing members of NP $\cap$ Co-NP exist.

REFERENCES

[1] A. Bulatov, A. Krokhin, and P. Jeavons, *Constraint satisfaction problems and finite algebras*, in Automata, Languages and Programming (Geneva, 2000), Lecture Notes in Comput. Sci. 1853, Springer-Verlag, Berlin, 2000, pp. 272–282.

[2] T. Feder and M. Vardi, *The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory*, SIAM J. Comput., 28 (1998), pp. 57–104.

[3] T. Gallai, *On directed paths and circuits*, in Theory of Graphs (Proc. Colloq., Tihany, 1966) Academic Press, New York, 1968, pp. 115–118.

[4] M. Hasse, *Zur algebraischen Begründung der Graphentheorie*. I., Math Nachr., 28 (1964/1965), pp. 275–290.

[5] P. Hell and J. Nešetřil, *On the complexity of H-coloring*, J. Combin. Theory Ser. B, 48 (1990), pp. 92–110.

[6] P. Hell and J. Nešetřil, *Graphs and Homomorphisms*, Oxford Lecture Ser. Math. Appl. 28, Oxford University Press, Oxford, UK, 2004.

[7] P. Hell, J. Nešetřil, and X. Zhu, *Duality and polynomial testing of tree homomorphisms*, Trans. Amer. Math. Soc., 348 (1996), pp. 1281–1297.

[8] P. Komárek, *Good Characterisations in the Class of Oriented Graphs*, Ph.D. thesis, Charles University, Prague, 1987 (in Czech).

[9] P. Komárek, *Some new good characterizations for directed graphs*, Časopis Pěst. Mat., 109 (1984), pp. 348–354.

[10] J. Nešetřil and A. Pultr, *On classes of relations and graphs determined by subobjects and factorobjects*, Discrete Math., 22 (1978), pp. 287–300.

[11] J. Nešetřil and C. Tardif, *Duality theorems for finite structures (characterizing gaps and good characterizations)*, J. Combin. Theory Ser. B, 80 (2000), pp. 80–97.

[12] J. Nešetřil and C. Tardif, *A Dualistic Approach to Bounding the Chromatic Number of a Graph*, ITI Series 2001-036, Institut Teoretické Informatiky, Charles University, Prague.

[13] J. Nešetřil and C. Tardif, *Density via duality*, Theoret. Comput. Sci., 287 (2002), pp. 585–591.

[14] B. Roy, *Nombre chromatique et plus longs chemins d'un graphe*, Rev. Francaise Informat. Recherche Opérationelle, 1 (1967), pp. 129–132.

[15] I. Švejdarová, *Colouring of Graphs and Dual Objects*, Ph.D. thesis, Charles University, Prague, 2003 (in Czech).

[16] L. M. Vitaver, *Determination of minimal coloring of vertices of a graph by means of Boolean powers of the incidence matrix*, Dokl. Akad. Nauk SSSR 147 (1962), pp. 758–759 (in Russian).

SIAM J. DISCRETE MATH.
Vol. 19, No. 4, pp. 921–937

AN APPLICATION OF RAMSEY THEORY TO CODING FOR THE OPTICAL CHANNEL[*]

NAVIN KASHYAP[†], PAUL H. SIEGEL[‡], AND ALEXANDER VARDY[‡]

Abstract. In this paper, we analyze bi-infinite sequences over the alphabet $\{0, 1, \ldots, q-1\}$, for an arbitrary $q \geq 2$, that satisfy the q-ary ghost pulse (qGP) constraint. A sequence $\mathbf{x} = (x_k)_{k\in\mathbb{Z}} \in \{0, 1, \ldots, q-1\}^{\mathbb{Z}}$ satisfies the qGP constraint if for all $k, l, m \in \mathbb{Z}$ such that x_k, x_l and x_m are nonzero and equal, x_{k+l-m} is also nonzero. This constraint arises in the context of coding for communication over a fiber optic medium. We show, using techniques from Ramsey theory, that if $\mathbf{x}$ satisfies the qGP constraint, then the set $\text{supp}(\mathbf{x}) = \{l \in \mathbb{Z} : \; x_l \neq 0\}$ is the disjoint union of cosets of some subgroup, $k\mathbb{Z}$, of $\mathbb{Z}$, and a set of zero density. We provide much sharper results in the special cases of $q = 2$ and $q = 3$. In the former case, we show that the corresponding binary ghost pulse constraint has zero capacity, and based on our results for the latter case, we conjecture that the capacity of the ternary ghost pulse constraint is also zero.

Key words. optical communication, constrained coding, ghost pulse constraints, Ramsey theory

AMS subject classifications. 94A55, 05D10

DOI. 10.1137/S089548010444585X

1. Introduction.

In this paper, we study the effect of a class of constraints which we call "ghost pulse" constraints imposed on sequences over a finite alphabet. Throughout the paper, we shall follow the standard convention of using $\mathbb{Z}$ to denote the set of all integers and $\mathbb{N}$ to denote the set of positive integers. Also, given $m, n \in \mathbb{Z}$, we shall take $[m, n]$ to be the set $\{k \in \mathbb{Z} : m \leq k \leq n\}$. Given an integer $q \geq 2$, let $\mathcal{A}_q = \{0, 1 \ldots, q-1\}$. For $\mathbf{x} = (x_k)_{k\in\mathbb{Z}} \in \mathcal{A}_q^{\mathbb{Z}}$, we define the support of $\mathbf{x}$ to be $\text{supp}(\mathbf{x}) = \{k \in \mathbb{Z} : x_k \neq 0\}$.

DEFINITION 1.1 (q-ary ghost pulse (qGP) constraint). *A sequence* $\mathbf{x} \in \mathcal{A}_q^{\mathbb{Z}}$ *satisfies the qGP constraint if for all* $k, l, m \in \text{supp}(\mathbf{x})$ *(*k, l, m *not necessarily distinct) such that* $x_k = x_l = x_m$, *we also have* $k + l - m \in \text{supp}(\mathbf{x})$.

We shall denote by $\mathcal{T}_q$ the set of all $\mathbf{x} \in \mathcal{A}_q^{\mathbb{Z}}$ that satisfy the qGP constraint. Furthermore, we shall use $\mathcal{S}_q$ to denote the set of all $\mathbf{y} \in \{0, 1\}^{\mathbb{Z}}$ such that there exists an $\mathbf{x} \in \mathcal{T}_q$ with $\text{supp}(\mathbf{x}) = \text{supp}(\mathbf{y})$. The object of this paper is to study the sequences in $\mathcal{S}_q$, particularly in the cases when q is 2 or 3. When $q = 2$, we refer to the corresponding constraint as the *binary ghost pulse (BGP)* constraint, and when $q = 3$, the corresponding constraint is called the *ternary ghost pulse (TGP)* constraint.

These ghost pulse constraints arise in the context of coding for communication over a fiber optic medium. In a typical optical communication scenario, a train of light pulses corresponding to a sequence of M bits is sent across the fiber optic medium that constitutes the optical channel. Each bit in the sequence is allocated a time

[*] Received by the editors September 8, 2004; accepted for publication (in revised form) April 17, 2005; published electronically December 30, 2005. This work was supported in part by the UC Discovery Grant Program, the Applied Micro Circuits Corporation, San Diego, CA, the National Science Foundation, and the David and Lucile Packard Foundation. The work was carried out while the first author was at the University of California, San Diego.

http://www.siam.org/journals/sidma/19-4/44585.html

[†] Department of Mathematics and Statistics, Queen's University, Kingston, ON, K7L 3N6, Canada (nkashyap@mast.queensu.ca).

[‡] Department of Electrical and Computer Engineering, University of California–San Diego, 9500 Gilman Drive, MC 0407, La Jolla, CA 92093-0407 (psiegel@ece.ucsd.edu, vardy@kilimanjaro.ucsd.edu).

slot of duration T, and a 1 or 0 is marked by the presence or absence of a pulse in that time slot. A nonlinear phenomenon known as four-wave mixing causes a transfer of energy from triples of pulses in "1" slots into certain "0" slots, creating spurious pulses called *ghost pulses*. It has been observed [1], [9] that the interaction of pulses in the kth, lth, and mth time slots (k, l, m need not all be distinct) in the pulse train pumps energy into the $(k+l-m)$th time slot. If this slot did not originally contain a pulse, i.e., if the $(k+l-m)$th bit was a 0 in the original M-bit sequence, then the transfer of energy creates a ghost pulse in the slot, thus changing the original 0 to a 1. The reader is referred to [5] for a more detailed description of this phenomenon.

The formation of ghost pulses may be modeled as follows: Let $b_0 b_1 \dots b_{M-1}$, $b_i \in \{0,1\}$, be the binary sequence corresponding to the transmitted train of pulses. If we have 1's in positions k, l, m (not necessarily all distinct) in this sequence, i.e., $b_k = b_l = b_m = 1$, and if $b_{k+l-m} = 0$, then the formation of a ghost pulse converts b_{k+l-m} to a 1. Note that if $b_0 b_1 \dots b_{M-1}$ were a subblock of a sequence $\mathbf{x} \in \mathcal{S}_2$, then no ghost pulses would be formed since if i is a position where a ghost pulse could potentially be created, then b_i is already a 1 by the definition of the BGP constraint. So, one way of eliminating the formation of ghost pulses when transmitting an arbitrary data sequence $b_0 b_1 \dots b_{M-1}$, $b_i \in \{0,1\}$, is to first encode the data sequence into a sequence $c_0 c_1 \dots c_{N-1}$ that is a subblock of some $\mathbf{x} \in \mathcal{S}_2$.

The efficiency of any coding scheme using subblocks of BGP-constrained sequences as codewords is limited by the *capacity*, $h(\mathcal{S}_2)$, of $\mathcal{S}_2$, which is defined as

$$h(\mathcal{S}_2) = \lim_{n \to \infty} \frac{\log_2 |\mathcal{B}_{2,n}|}{n}, \tag{1}$$

where $\mathcal{B}_{2,n}$ denotes the set of all length-n subblocks of sequences in $\mathcal{S}_2$. The closer $h(\mathcal{S}_2)$ is to 1, the more efficient are the coding schemes based on BGP-constrained sequences. However, it is easily shown that $h(\mathcal{S}_2) = 0$ as a consequence of the following simple characterization of sequences in $\mathcal{S}_2$.

THEOREM 1.2. *A binary sequence* $\mathbf{x}$ *is in* $\mathcal{S}_2$ *if and only if* $\operatorname{supp}(\mathbf{x}) = \emptyset$ *or* $\operatorname{supp}(\mathbf{x}) = a + k\mathbb{Z}$ *for some* $a, k \in \mathbb{Z}$.

Proof. It is clear from the definition of the BGP constraint that if $\mathbf{x} \in \{0,1\}^{\mathbb{Z}}$ is such that $\operatorname{supp}(\mathbf{x}) = \emptyset$ or $\operatorname{supp}(\mathbf{x}) = a + k\mathbb{Z}$, then $\mathbf{x} \in \mathcal{S}_2$. For the converse, suppose that $\mathbf{x} \in \mathcal{S}_2$ is such that $\operatorname{supp}(\mathbf{x}) \neq \emptyset$. Take any $a \in \operatorname{supp}(\mathbf{x})$ and let $H = \operatorname{supp}(\mathbf{x}) - a = \{k - a : x_k \neq 0\}$. It is easily verified that H is a subgroup of $\mathbb{Z}$, and hence, $H = k\mathbb{Z}$ for some integer k. Thus, $\operatorname{supp}(\mathbf{x}) = a + H = a + k\mathbb{Z}$. □

COROLLARY 1.3. $h(\mathcal{S}_2) = 0$.

Proof. It follows from the above theorem that $|\mathcal{B}_{2,n}| = O(n^2)$, which implies that $h(\mathcal{S}_2) = 0$. □

Thus, any coding scheme based on BGP-constrained sequences is bound to be inefficient in terms of rate. So, we need to consider alternative approaches to dealing with the ghost pulse problem.

One approach that has been suggested to mitigate the formation of ghost pulses is to apply, at the transmitter end, a phase shift of π to some of the pulses in the "1" time slots [7], [2]. The interaction of pulses with different phases suppresses the formation of ghost pulses at certain locations due to destructive interference. However, three pulses with the same phase can still interact to create ghost pulses. We can effectively think of this phase modulation technique as converting a binary sequence $b_0 b_1 \dots b_{N-1}$ into a ternary sequence $c_0 c_1 \dots c_{N-1}$, with $c_i \in \{-1, 0, 1\}$, such that $b_i = |c_i|$ for all $i \in \{0, 1, \dots, N-1\}$. As a first-order approximation of the true situation, we shall assume that the only case in which a ghost pulse is formed is when we have $c_k = c_l = c_m = 1$ or $c_k = c_l = c_m = -1$, and $c_{k+l-m} = 0$.

Now, if $\mathbf{x} \in \{-1, 0, 1\}^{\mathbb{Z}}$ is a sequence satisfying the TGP constraint[1] and if $c_0 c_1 \ldots c_{N-1}$ is a subblock of $\mathbf{x}$, then by the first-order approximation stated above, $c_0 c_1 \ldots c_{N-1}$ can be transmitted without error across the optical channel. Thus, finite-length subblocks of sequences in $\mathcal{T}_3$ can be used as codewords for encoding binary data sequences.

However, there is a catch. In reality, an optical receiver can detect only the amplitude of the optical signal at the channel output, not its phase. What this means is that if the transmitted ternary sequence was $c_0 c_1 \ldots c_{N-1}$, then the receiver sees only the sequence $|c_0|, |c_1|, \ldots, |c_{N-1}|$; i.e., the receiver cannot distinguish a 1 from a -1. As a result, we cannot use two ternary sequences that differ only in phase (i.e., only in sign) to encode two different binary data sequences.

So, the proper procedure to encode and transmit a finite-length binary data sequence $a_0 a_1 \ldots a_{M-1}$ is to first encode it with a subblock $b_0 b_1 \ldots b_{N-1}$ of some sequence in $\mathcal{S}_3$ which, before transmission, is converted to a subblock, $c_0 c_1 \ldots c_{N-1}$, of some sequence in $\mathcal{T}_3$. At the channel output, the receiver detects the sequence $b_0 b_1 \ldots b_{N-1}$ which can be decoded correctly to recover $a_0 a_1 \ldots a_{M-1}$. We thus have a rather unusual coding problem because even though the sequence being transmitted is a ternary sequence, the alphabet used for the encoding of information is effectively binary.

Consequently, the efficiency of any coding scheme that uses TGP-constrained sequences is limited by the capacity, $h(\mathcal{S}_3)$, of the set $\mathcal{S}_3$, which is defined analogously to (1) as follows:

$$h(\mathcal{S}_3) = \lim_{n \to \infty} \frac{\log_2 |\mathcal{B}_{3,n}|}{n}, \tag{2}$$

where $\mathcal{B}_{3,n}$ denotes the set of all length-n subblocks of sequences in $\mathcal{S}_3$. It should be pointed out that the existence of the limits in (1) and (2) follows by standard arguments from the following fact (cf. [6, Chapter 4]): If $a_1, a_2, \ldots$ is a sequence of nonnegative numbers such that $a_{m+n} \leq a_m + a_n$ for all $m, n \geq 1$, then $\lim_{n\to\infty} a_n/n$ exists and equals $\inf_{n \geq 1} a_n/n$.

In this paper, we analyze the structure of the sequences in $\mathcal{S}_3$ in an attempt to provide a simple characterization for them along the lines of Theorem 1.2, which could then be used to determine $h(\mathcal{S}_3)$. Unfortunately, the TGP constraint is much harder to analyze than its binary counterpart. It is actually instructive to study the qGP constraint for arbitrary $q \geq 2$ as it provides useful insight into the ternary case. In fact, extension to the q-ary alphabet allows for an unexpectedly simple and elegant analysis based on results drawn from the branch of mathematics known as Ramsey theory.

Using results from Ramsey theory, we show in Theorem 3.1 that any sequence $\mathbf{y} \in \mathcal{S}_q$ is "almost periodic" in the sense that it can be transformed into a periodic sequence by changing a relatively sparse subset of the 1's to 0's. More precisely, we show that if $\mathbf{y} \in \mathcal{S}_q$, then there exists a subset $N(\mathbf{y}) \subset \operatorname{supp}(\mathbf{y})$ such that $N(\mathbf{y})$ has density[2] 0 and $\operatorname{supp}(\mathbf{y}) \setminus N(\mathbf{y})$ is a union of cosets of some subgroup, $k\mathbb{Z}$, of $\mathbb{Z}$. For sequences in $\mathcal{S}_3$, we make this result much stronger by showing in Theorem 4.4 that

[1] In Definition 1.1, ternary sequences are defined over the alphabet $\{0, 1, 2\}$. We simply identify the symbol 2 with -1.

[2] Density is defined in section 2.

any $\mathbf{y} \in \mathcal{S}_3$ can be made periodic by changing at most two 1's to 0's. In fact, this theorem provides a simple and complete description of the aperiodic sequences in $\mathcal{S}_3$. We also provide a useful characterization (Theorem 4.1) of periodic sequences in $\mathcal{S}_3$, which we use to completely describe all such sequences of prime period (Theorem 4.3). Based on these results and some numerical evidence, we conjecture that $h(\mathcal{S}_3) = 0$.

The remainder of the paper is organized as follows. In section 2, we provide the background from Ramsey theory needed for our proofs. Section 3 contains our analysis of qGP-constrained sequences, and section 4 presents the analysis for TGP-constrained sequences. In section 5, we present some numerical evidence in support of our conjecture that $h(\mathcal{S}_3) = 0$.

2. Some Ramsey theory. Given a set I and a positive integer k, we refer to any function $\chi : I \to [1, k]$ as a *k-coloring* of I. Observe that if $V_j = \{i \in I : \chi(i) = j\}$, then the sets V_j, $j = 1, 2, \dots , k$, form a partition of I, which we shall call the *chromatic partition* (with respect to the coloring χ) of I. The sets V_j are often called the *color classes* of χ. A subset $J \subset I$ is said to be *monochromatic* (wrt χ) if $J \subset V_j$ for some $j \in [1, k]$.

Ramsey theory is a branch of combinatorics that deals with structure which is preserved under partitions [3]. A typical result from Ramsey theory guarantees that when some set I is finitely colored, then some structure of the set I appears in monochromatic form. One of the classic results of Ramsey theory is the following theorem due to Schur [4, Chapter 3, Theorem 1].

Theorem 2.1 (Schur's theorem). *Given a $k \in \mathbb{N}$, there exists an $N(k) \in \mathbb{N}$ such that for all $n \geq N(k)$ every k-coloring of $[1, n]$ contains a monochromatic solution to $x + y = z$.*

To put it another way, Schur's theorem states that given a $k \in \mathbb{N}$, for all sufficiently large n, if we partition $[1, n]$ into k subsets, $V_1, V_2 \dots , V_k$, then there exist $x, y, z \in V_i$ for some i that satisfy $x + y = z$. The smallest integer $N(k)$ for which the statement of Schur's theorem holds is referred to as the kth Schur number and is denoted by $S(k)$. The exact value of $S(k)$ is known only for $k = 1, 2, 3, 4$: $S(1) = 2$, $S(2) = 5$, $S(3) = 14$, $S(4) = 45$ [8, Sequence A030126].

Another well-known result from Ramsey theory, known as van der Waerden's theorem [4, Chapter 2, Theorem 1], guarantees the existence of arbitrarily long monochromatic arithmetic progressions in any k-coloring of the integers. Recall that an arithmetic progression (AP) of length l is a subset of the integers of the form $\{a + id : i = 0, 1, \dots , l - 1\}$ for some $a \in \mathbb{Z}$ and $d \in \mathbb{N}$. We shall require a stronger form of van der Waerden's theorem, for which we need the following definition.

Definition 2.2 (upper density). *Given $I \subset \mathbb{Z}$, the upper density of I is defined to be*

$$\overline{d}(I) = \limsup_{n\to\infty} \frac{|I \cap [-n, n]|}{2n + 1}.$$

Note that if $I_1, I_2, \dots , I_k$ form a (finite) partition of $I \subset \mathbb{Z}$, then $\overline{d}(I) = \sum_{i=1}^{k} \overline{d}(I_i)$. In particular, if χ is a k-coloring of $\mathbb{Z}$ and $\{V_1, V_2, \dots , V_k\}$ is the corresponding chromatic partition of $\mathbb{Z}$, then $\sum_{j=1}^{k} \overline{d}(V_j) = 1$ since $\overline{d}(\mathbb{Z}) = 1$. Therefore, in any k-coloring, χ, of $\mathbb{Z}$, at least one of the color classes, V_j, $j = 1, 2, \dots , k$, of χ must have positive upper density. Thus, van der Waerden's theorem is a consequence of the following stronger result, known as Szemerédi's theorem [4, Chapter 2, p. 43].

Theorem 2.3 (Szemerédi's theorem). *If $I \subset \mathbb{Z}$ has positive upper density (i.e., $\overline{d}(I) > 0$), then for any $l \in \mathbb{N}$, I contains an AP of length l.*

The relevance of colorings to the study of qGP-constrained sequences can be seen from the following simple lemma.

Lemma 2.4. *A binary sequence $\mathbf{y}$ is in $\mathcal{S}_q$ if and only if there exists a $(q-1)$-coloring, χ, of $\operatorname{supp}(\mathbf{y})$ such that whenever $k, l, m \in \operatorname{supp}(\mathbf{y})$ satisfy $\chi(k) = \chi(l) = \chi(m)$, then $k + l - m \in \operatorname{supp}(\mathbf{y})$.*

Proof. If $\mathbf{y}$ is in $\mathcal{S}_q$, then there exists an $\mathbf{x} \in \mathcal{T}_q$ with $\operatorname{supp}(\mathbf{x}) = \operatorname{supp}(\mathbf{y})$. For $k \in \operatorname{supp}(\mathbf{y})$, let $\chi(k) = x_k$. Then, by definition of the qGP constraint, χ is a $(q-1)$-coloring of $\operatorname{supp}(\mathbf{y})$ with the required property.

Conversely, if $\mathbf{y} \in \{0,1\}^{\mathbb{Z}}$ is such that χ is a $(q-1)$-coloring of $\operatorname{supp}(\mathbf{y})$ as in the statement of the lemma, then let $\mathbf{x} = (x_k)_{k\in\mathbb{Z}}$ be the sequence defined by $x_k = 0$ if $k \notin \operatorname{supp}(\mathbf{y})$ and $x_k = \chi(k)$ if $k \in \operatorname{supp}(\mathbf{y})$. Thus, $\operatorname{supp}(\mathbf{x}) = \operatorname{supp}(\mathbf{y})$, and by definition of the qGP constraint, $\mathbf{x} \in \mathcal{T}_q$. □

3. The qGP constraint.

We shall use the results from Ramsey theory provided in the previous section to prove our main result on qGP-constrained sequences, which we state next.

Theorem 3.1. *For $q \geq 2$, if $\mathbf{x} \in \mathcal{S}_q$ or $\mathbf{x} \in \mathcal{T}_q$, then there exists an integer $k \geq 0$ and a set $I \subset [0, k-1]$, both depending on $\mathbf{x}$, such that*

$$\bigcup_{i\in I}(k\mathbb{Z} + i) \subset \operatorname{supp}(\mathbf{x})$$

and

$$\overline{d}\left(\operatorname{supp}(\mathbf{x}) \setminus \bigcup_{i\in I}(k\mathbb{Z} + i)\right) = 0.$$

In other words, outside a set of density 0, $\operatorname{supp}(\mathbf{x})$ is a union of cosets of some subgroup, $k\mathbb{Z}$, of $\mathbb{Z}$. It is enough to prove this theorem for $\mathbf{x} \in \mathcal{T}_q$ since for any $\mathbf{y} \in \mathcal{S}_q$, there exists an $\mathbf{x} \in \mathcal{T}_q$ with $\operatorname{supp}(\mathbf{x}) = \operatorname{supp}(\mathbf{y})$. Our proof of the theorem relies on the following proposition, which shows that if $\mathbf{x} \in \mathcal{T}_q$ is such that $\operatorname{supp}(\mathbf{x})$ contains a sufficiently large number of consecutive terms of $a + k\mathbb{Z}$ for some $a, k \in \mathbb{Z}$, then it must in fact contain all of $a + k\mathbb{Z}$. Recall that for $q \in \mathbb{N}$, $S(q)$ is the qth Schur number.

Proposition 3.2. *For $q \geq 2$, if $\mathbf{x} \in \mathcal{T}_q$ is such that $\operatorname{supp}(\mathbf{x})$ contains an $S(q-1)$-term AP, $\{a + jk : 1 \leq j \leq S(q-1)\}$ for some $a, k \in \mathbb{Z}$, then $a + k\mathbb{Z} \subset \operatorname{supp}(\mathbf{x})$.*

Proof. Suppose that $\mathbf{x} = (x_m)_{m\in\mathbb{Z}} \in \mathcal{T}_q$ is such that $\operatorname{supp}(\mathbf{x})$ contains $a + jk$, $1 \leq j \leq S(q-1)$. We shall show that a and $a + (S(q-1)+1)k$ are also in $\operatorname{supp}(\mathbf{x})$ so that the result then follows by induction.

Define a $(q-1)$-coloring, χ, of $[1, S(q-1)]$ via $\chi(j) = x_{a+jk}$ for $j \in [1, S(q-1)]$. This is indeed a $(q-1)$-coloring since $a + jk \in \operatorname{supp}(\mathbf{x})$, and hence, $x_{a+jk} \neq 0$ for $j \in [1, S(q-1)]$. By Schur's theorem, there exist $r, s, t \in [1, S(q-1)]$ such that $r + s = t$ and $\chi(r) = \chi(s) = \chi(t)$ or, equivalently, $x_{a+rk} = x_{a+sk} = x_{a+tk}$. But now, $(a + rk) + (a + sk) - (a + tk) = a$ so that, by definition of the qGP constraint, $a \in \operatorname{supp}(\mathbf{x})$ as well.

A similar argument using the coloring of $[1, S(q-1)]$ defined by

$$\hat{\chi}(j) = \chi(S(q-1) + 1 - j) = x_{a+(S(q-1)+1-j)k}$$

proves that $a + (S(q-1)+1)k$ is also in $\operatorname{supp}(\mathbf{x})$, which completes the proof of the proposition. □

For $\mathbf{x} \in \mathcal{A}_q^{\mathbb{Z}}$ and $j \in [0, q-1]$, we define

$$V_j(\mathbf{x}) = \{k \in \mathbb{Z} : x_k = j\}. \tag{3}$$

Thus, the sets $V_j(\mathbf{x})$, $j \in [0, q-1]$, constitute a partition of $\mathbb{Z}$, while $V_j(\mathbf{x})$, $j \in [1, q-1]$, is a partition of $\mathrm{supp}(\mathbf{x})$. Note that $\mathbf{x} \in \mathcal{T}_q$ if and only if the sets $V_j(\mathbf{x})$ satisfy the following condition: For any $j \in [1, q-1]$, if $k, l, m \in V_j(\mathbf{x})$, then $k + l - m \in \mathrm{supp}(\mathbf{x})$. The following is a useful corollary to the above proposition.

Corollary 3.3. *Let $q \geq 2$ and $N = \lceil \frac{S(q-1)+2}{3} \rceil$. If $\mathbf{x} \in \mathcal{T}_q$ is such that, for some $j \in [1, q-1]$, $V_j(\mathbf{x})$ contains an N-term AP $\{a + \ell k : 0 \leq \ell \leq N-1\}$ for some $a, k \in \mathbb{Z}$, then $a + k\mathbb{Z} \subset \mathrm{supp}(\mathbf{x})$.*

Proof. Suppose that $a, k \in \mathbb{Z}$ are such that $\{a + \ell k : 0 \leq \ell \leq N-1\} \subset V_j(\mathbf{x})$ for some $j \in [1, q-1]$. Note that for $0 \leq \ell \leq N-2$, $a + (N+\ell)k = (a + (N-1)k) + (a + (N-1)k) - (a + (N-2-\ell)k)$. Since $a + (N-1)k, a + (N-2-\ell)k \in V_j(\mathbf{x})$, it follows from the definition of the qGP constraint that $a + (N+\ell)k \in \mathrm{supp}(\mathbf{x})$.

Similarly, for $1 \leq \ell \leq N-1$, $a - \ell k = a + a - (a + \ell k)$, and hence, $a - \ell k \in \mathrm{supp}(\mathbf{x})$ as well. Thus, $\mathrm{supp}(\mathbf{x})$ contains the $(3N+2)$-term AP $\{a + \ell k : -(N-1) \leq \ell \leq 2(N-1)\}$. Since $3N + 2 \geq S(q-1)$, the result follows from Proposition 3.2. ◻

The next lemma forms the crux of the proof of Theorem 3.1.

Lemma 3.4. *For $\mathbf{x} \in \mathcal{T}_q$, if $\overline{d}(V_j(\mathbf{x})) > 0$ for some $j \in [1, q-1]$, then there exists a $k_j \in \mathbb{N}$ such that if we let*

$$I_j = \{i \in [0, k_j - 1] : \ |V_j(\mathbf{x}) \cap (k_j\mathbb{Z} + i)| > 0\},$$

then

$$V_j(\mathbf{x}) \subset \bigcup_{i \in I_j} (k_j\mathbb{Z} + i) \subset \mathrm{supp}(\mathbf{x}).$$

Proof. By the definition of I_j, it is obvious that for any j, $V_j \subset \bigcup_{i \in I_j} (k_j\mathbb{Z} + i)$. So, we shall show that if $\overline{d}(V_j(\mathbf{x})) > 0$ for some $j \in [1, q-1]$, then there exists $k_j \neq 0$ such that, with I_j as defined above, $\bigcup_{i \in I_j} (k_j\mathbb{Z} + i) \subset \mathrm{supp}(\mathbf{x})$. Indeed, it suffices to show that for each $i \in I_j$, $k_j\mathbb{Z} + i \subset \mathrm{supp}(\mathbf{x})$.

Without loss of generality, we may assume that $\overline{d}(V_1(\mathbf{x})) > 0$. By Szemerédi's theorem, $V_1(\mathbf{x})$ contains an $S(q-1)$-term AP $\{a + jk_1 : 1 \leq j \leq S(q-1)\}$ for some $a \in \mathbb{Z}$ and $k_1 \in \mathbb{N}$. Now, take any $i \in I_1$, where I_1 is as in the statement of the lemma. We need to show that $k_1\mathbb{Z} + i \subset \mathrm{supp}(\mathbf{x})$. Since i is in I_1, there exists an $m \in \mathbb{Z}$ such that $i + mk_1 \in V_1(\mathbf{x})$. But now, for any $j \in [1, S(q-1)]$, since $a + jk_1 \in V_1(\mathbf{x})$, the qGP constraint implies that $(i + mk_1) + (a + jk_1) - (a + k_1) = i + (m + j - 1)k_1$ is in $\mathrm{supp}(\mathbf{x})$. We thus have an $S(q-1)$-term AP $\{i + (m + j - 1)k_1 : 1 \leq j \leq S(q-1)\}$ in $\mathrm{supp}(\mathbf{x})$, and hence by Proposition 3.2, $i + k_1\mathbb{Z} \subset \mathrm{supp}(\mathbf{x})$, as desired. ◻

We are now in a position to prove Theorem 3.1. Given $\mathbf{x} \in \mathcal{T}_q$, we shall let $J_1 = \{j \in [1, q-1] : \overline{d}(V_j(\mathbf{x})) > 0\}$ and $J_2 = \{j \in [1, q-1] : \overline{d}(V_j(\mathbf{x})) = 0\}$. Also, let $P(\mathbf{x}) = \bigcup_{j \in J_1} V_j(\mathbf{x})$ and $N(\mathbf{x}) = \bigcup_{j \in J_2} V_j(\mathbf{x})$. Clearly, $P(\mathbf{x}), N(\mathbf{x})$ form a partition of $\mathrm{supp}(\mathbf{x})$ with $\overline{d}(P(\mathbf{x})) > 0$, if $P(\mathbf{x}) \neq \emptyset$, and $\overline{d}(N(\mathbf{x})) = 0$.

Proof of Theorem 3.1. As mentioned earlier, it suffices to prove the theorem for $\mathbf{x} \in \mathcal{T}_q$. If $\overline{d}(\mathrm{supp}(\mathbf{x})) = 0$, then we may take $k = 0$. So, we may assume that $\overline{d}(\mathrm{supp}(\mathbf{x})) > 0$ so that $J_1 \neq \emptyset$. For each $j \in J_1$, let k_j and I_j be as in the statement of Lemma 3.4, and let $k = \mathrm{lcm}\{k_j : j \in J_1\}$ be the least common multiple of the k_j's.

Since

$$k_j\mathbb{Z} + i = \bigcup_{\ell=0}^{k/k_j - 1} (k\mathbb{Z} + \ell k_j + i),$$

if we define $\widehat{I}_j$ to be $\{\ell k_j + i : i \in I_j, \ell \in [0, k/k_j - 1]\}$, then

$$\bigcup_{i \in I_j} (k_j\mathbb{Z} + i) = \bigcup_{i \in \widehat{I}_j} (k\mathbb{Z} + i).$$

Therefore, for each $j \in J_1$,

$$V_j(\mathbf{x}) \subset \bigcup_{i \in \widehat{I}_j} (k_j\mathbb{Z} + i) \subset \mathrm{supp}(\mathbf{x}).$$

Now, taking $I = \bigcup_{j \in J_1} \widehat{I}_j$, we see that

$$P(\mathbf{x}) \subset \bigcup_{i \in I} (k\mathbb{Z} + i) \subset \mathrm{supp}(\mathbf{x}).$$

Finally,

$$\mathrm{supp}(\mathbf{x}) \setminus \bigcup_{i \in I} (k\mathbb{Z} + i) \subset \mathrm{supp}(\mathbf{x}) \setminus P(\mathbf{x}) = N(\mathbf{x})$$

from which it follows that

$$\overline{d}\left(\mathrm{supp}(\mathbf{x}) \setminus \bigcup_{i \in I} (k\mathbb{Z} + i)\right) = 0,$$

which completes the proof of the theorem. □

It is straightforward to see that the set I in the above proof is in fact the set $\{i \in [0, k-1] : |P(\mathbf{x}) \cap (k\mathbb{Z} + i)| > 0\}$.

COROLLARY 3.5. *If* $\mathbf{x} \in \mathcal{T}_q$ *is such that* $N(\mathbf{x}) = \emptyset$, *then the sequence* $\mathbf{y} \in \mathcal{S}_q$ *with* $\mathrm{supp}(\mathbf{y}) = \mathrm{supp}(\mathbf{x})$ *is a periodic sequence.*

Proof. If $\mathrm{supp}(\mathbf{x}) = \emptyset$, then $\mathbf{x}$, as well as the corresponding $\mathbf{y} \in \mathcal{S}_q$, is simply the all-zeros sequence, $0^{\mathbb{Z}}$, which is periodic. If $\mathrm{supp}(\mathbf{x}) \neq \emptyset$ but $N(\mathbf{x}) = \emptyset$, then $\mathrm{supp}(\mathbf{x}) = P(\mathbf{x})$. So, as in the proof of the above theorem, there exists a $k \in \mathbb{N}$ and an $I \subset [0, k-1]$ such that

$$\mathrm{supp}(\mathbf{x}) \subset \bigcup_{i \in I} (k\mathbb{Z} + i) \subset \mathrm{supp}(\mathbf{x}).$$

Hence, $\mathrm{supp}(\mathbf{x}) = \bigcup_{i \in I} (k\mathbb{Z} + i)$. It now follows that the corresponding $\mathbf{y} \in \mathcal{S}_q$ is periodic with period k. □

It appears to be difficult to strengthen Theorem 3.1 any further, for example, to give a complete description of the sequences that are allowed to be in $\mathcal{T}_q$ or $\mathcal{S}_q$ for arbitrary q. However, for $q = 3$, which is our main case of interest, we can do much better than Theorem 3.1, as we show in the next section.

4. The TGP constraint. In this section, we provide a means of characterizing the binary sequences that are in $\mathcal{S}_3$. Separate characterizations are provided for binary sequences that are periodic and for those that are not. Recall that $\mathbf{y} = (y_m)_{m\in\mathbb{Z}} \in \{0,1\}^{\mathbb{Z}}$ is *periodic* if there exists a $k \in \mathbb{N}$ such that $y_m = y_{m+k}$ for all $m \in \mathbb{Z}$. The integer k is referred to as a *period* of $\mathbf{y}$. The *fundamental period* of a periodic sequence, $\mathbf{y}$, is the smallest $k \in \mathbb{N}$ that is a period of $\mathbf{y}$. Note that if $\mathbf{y} \in \{0,1\}^{\mathbb{Z}}$ is not the all-zeros sequence $0^{\mathbb{Z}}$, then $\mathbf{y}$ is periodic if and only if there exists a $k \in \mathbb{N}$ and a nonempty $I \subset [0, k-1]$ such that $\operatorname{supp}(\mathbf{y}) = \bigcup_{i\in I}(k\mathbb{Z} + i)$. The following theorem shows that a nonzero periodic sequence having k as a period is in $\mathcal{S}_3$ if and only if it satisfies a certain "modulo-k" TGP constraint, in a manner made precise in the statement of the theorem.

THEOREM 4.1. *Let $\mathbf{y} \in \{0,1\}^{\mathbb{Z}}$, $\mathbf{y} \neq 0^{\mathbb{Z}}$, be periodic so that there exist a $k \in \mathbb{N}$ and a nonempty $I \subset [0, k-1]$ such that $\operatorname{supp}(\mathbf{y}) = \bigcup_{i\in I}(k\mathbb{Z} + i)$. Then, $\mathbf{y} \in \mathcal{S}_3$ if and only if there exists a 2-coloring, χ, of I such that whenever $i_1, i_2, i_3 \in I$ satisfy $\chi(i_1) = \chi(i_2) = \chi(i_3)$, then $i_1 + i_2 - i_3 \mod k \in I$.*

Since $I \equiv \operatorname{supp}(\mathbf{y}) \pmod{k}$, a comparison of the statement of the above theorem with that of Lemma 2.4 highlights the "modulo-k" nature of the TGP constraint imposed on periodic sequences in $\mathcal{S}_3$. The modulo-k connection can be made explicit in terms of the definition given below. For $k \in \mathbb{N}$, let $\mathbb{Z}/k$ denote the group of integers modulo k; i.e., $\mathbb{Z}/k$ is the set $[0, k-1]$ equipped with the operation of modulo-k addition.

DEFINITION 4.2 (TGP-coloring). *For $k \in \mathbb{N}$ and $I \subset \mathbb{Z}/k$, a TGP-coloring of I is a 2-coloring, χ, of I such that whenever $i_1, i_2, i_3 \in I$ satisfy $\chi(i_1) = \chi(i_2) = \chi(i_3)$, then $i_1 + i_2 - i_3 \in I$.*

We would like to clarify that whenever we write $I \subset \mathbb{Z}/k$, we tacitly assume that I gets equipped with the same operation as $\mathbb{Z}/k$. So, in the above definition, $i_1 + i_2 - i_3$ is in fact taken modulo k.

A subset $I \subset \mathbb{Z}/k$ is said to be *TGP-colorable* if there exists a TGP-coloring of I. Thus, we may restate Theorem 4.1 as follows: Let $\mathbf{y} \in \{0,1\}^{\mathbb{Z}}$, $\mathbf{y} \neq 0^{\mathbb{Z}}$, be periodic so that there exist a $k \in \mathbb{N}$ and a nonempty $I \subset [0, k-1]$ such that $\operatorname{supp}(\mathbf{y}) = \bigcup_{i\in I}(k\mathbb{Z} + i)$. Then, $\mathbf{y} \in \mathcal{S}_3$ if and only if $I \subset \mathbb{Z}/k$ is TGP-colorable.

While the $\mathbb{Z}/k$ TGP-colorability condition is easier to check than the full-blown TGP condition, it is still very hard in practice to verify that this condition holds for an arbitrary $I \subset \mathbb{Z}/k$. However, if p is a prime, then we can determine precisely which subsets of $\mathbb{Z}/p$ are TGP-colorable.

THEOREM 4.3. *Let p be prime. Then, $I \subset \mathbb{Z}/p$ is TGP-colorable if and only if one of the following holds:*

(i) $|I| \leq 2$*;*

(ii) $I = [0, p-1]$*;*

(iii) $p = 5$ *and* $|I| = 4$.

For nonprime k, the problem of determining all the subsets of $\mathbb{Z}/k$ that are TGP-colorable remains open. In other words, we do not yet have an easily verifiable characterization for periodic sequences in $\mathcal{S}_3$ whose fundamental period is nonprime. In Table 5.1 in the next section, we list the number of TGP-colorable subsets of $\mathbb{Z}/k$ for nonprime $k \leq 20$, obtained by means of an exhaustive computer search.

Luckily, the problem of determining which *aperiodic* sequences are in $\mathcal{S}_3$ turns out to be a lot easier. There is a simple characterization of such sequences, which is presented in the following theorem.

THEOREM 4.4. *Let $\mathbf{y} \in \{0,1\}^{\mathbb{Z}}$ be an aperiodic sequence. Then, $\mathbf{y} \in \mathcal{S}_3$ if and* *only if one of the following conditions holds:*

(i) $1 \le |\operatorname{supp}(\mathbf{y})| \le 2$*;*

(ii) *there exist a* $k \in \mathbb{N}$ *and an* $i \in [0, k-1]$ *such that* $\operatorname{supp}(\mathbf{y}) = (k\mathbb{Z} + i) \cup V$*, with* $V = \{j\}$ *for some* $j \in \mathbb{Z}$*,* $j \not\equiv i \pmod{k}$*;*

(iii) *there exist a* $t \in \mathbb{N}$ *and an* $i \in [0, 3t-1]$ *such that* $\operatorname{supp}(\mathbf{y}) = (3t\mathbb{Z} + i) \cup V$*, with* $|V| = 2$ *and* $V \equiv \{t+i, 2t+i\} \pmod{3t}$*.*

The remainder of this section is devoted to the proofs of Theorems 4.1, 4.3, and 4.4. For the purpose of the proofs, we shall find it convenient to define the function $\pi : \{0,1,2\}^{\mathbb{Z}} \to \{0,1\}^{\mathbb{Z}}$ as follows: for $\mathbf{x} \in \{0,1,2\}^{\mathbb{Z}}$, $\pi(\mathbf{x})$ is the unique $\mathbf{y} \in \{0,1\}^{\mathbb{Z}}$ such that $\operatorname{supp}(\mathbf{y}) = \operatorname{supp}(\mathbf{x})$. Observe that $\pi(\mathcal{T}_3) = \mathcal{S}_3$.

4.1. Periodic sequences in $\mathcal{S}_3$.

The proof of Theorem 4.1 is based on the following lemma.

LEMMA 4.5. *Let* $\mathbf{y} \in \mathcal{S}_3$ *be such that* $\operatorname{supp}(\mathbf{y}) = \bigcup_{i \in I}(k\mathbb{Z} + i)$ *for some* $k \in \mathbb{N}$ *and a nonempty* $I \subset [0, k-1]$*. Then, there exists an* $\mathbf{x} \in \mathcal{T}_3$ *such that* $\pi(\mathbf{x}) = \mathbf{y}$*,* $V_1(\mathbf{x}) = \bigcup_{i \in I_1}(k\mathbb{Z} + i)$*, and* $V_2(\mathbf{x}) = \bigcup_{i \in I_2}(k\mathbb{Z} + i)$ *for some partition* $\{I_1, I_2\}$ *of* I*.*

Proof. Let $\mathbf{y}$, k, and I be as in the statement of the lemma, and let $\overline{I} = [0, k-1] \setminus I$. Since $\mathbf{y} \in \mathcal{S}_3$, there exists a $\mathbf{z} \in \mathcal{T}_3$ such that $\pi(\mathbf{z}) = \mathbf{y}$. For each $i \in I$, $k\mathbb{Z} + i \subset \operatorname{supp}(\mathbf{z}) = V_1(\mathbf{z}) \cup V_2(\mathbf{z})$, where $V_1(\mathbf{z}), V_2(\mathbf{z})$ are as defined in (3). Let $I_1 = \{i \in I : |(kZ + i) \cap V_1(\mathbf{z})| > 0\}$, and let $I_2 = I \setminus I_1$. Thus, $\{I_1, I_2\}$ is a partition of I. Now, define $\mathbf{x} = (x_j)_{j \in \mathbb{Z}} \in \{0,1,2\}^{\mathbb{Z}}$ as follows:

$$x_j = \begin{cases} 0 & \forall\, j \in k\mathbb{Z} + i,\ i \notin I, \\ 1 & \forall\, j \in k\mathbb{Z} + i,\ i \in I_1, \\ 2 & \forall\, j \in k\mathbb{Z} + i,\ i \in I_2. \end{cases}$$

Clearly, $\pi(\mathbf{x}) = \mathbf{y}$, $V_1(\mathbf{x}) = \bigcup_{i \in I_1}(k\mathbb{Z} + i)$, and $V_2(\mathbf{x}) = \bigcup_{i \in I_2}(k\mathbb{Z} + i)$. We shall show that $\mathbf{x} \in \mathcal{T}_3$.

Suppose, to the contrary, that $\mathbf{x} \notin \mathcal{T}_3$ so that there exist $p, q, r \in V_1(\mathbf{x})$ or $p, q, r \in V_2(\mathbf{x})$ such that $p + q - r \notin \operatorname{supp}(\mathbf{x})$. Without loss of generality, we may assume that $p, q, r \in V_1(\mathbf{x})$ and $p + q - r \notin \operatorname{supp}(\mathbf{x}) = \bigcup_{i \in I}(k\mathbb{Z} + i)$. Thus, $p + q - r \in \bigcup_{i \in \overline{I}}(k\mathbb{Z} + i)$, so that $p + q - r \mod k \notin I$.

Since $p, q, r \in V_1(x)$, $p \equiv i_1 \pmod{k}$, $q \equiv i_2 \pmod{k}$, and $r \equiv i_3 \pmod{k}$ for some $i_1, i_2, i_3 \in I_1$. Now, by definition of I_1, for each $i \in I_1$, there exists $t \in V_1(\mathbf{z})$ such that $t \equiv i \pmod{k}$. In particular, there exist $p', q', r' \in V_1(\mathbf{z})$ such that $p' \equiv i_1 \pmod{k}$, $q' \equiv i_2 \pmod{k}$, and $r' \equiv i_3 \pmod{k}$. In other words, $p' \equiv p \pmod{k}$, $q' \equiv q \pmod{k}$, and $r' \equiv r \pmod{k}$. Now, since $\mathbf{z} \in \mathcal{T}_3$, $p' + q' - r' \in \operatorname{supp}(\mathbf{z}) = \bigcup_{i \in I}(k\mathbb{Z} + i)$, and hence, $p' + q' - r' \mod k \in I$. But since $p' + q' - r' \equiv p + q - r \pmod{k}$, this contradicts $p + q - r \mod k \notin I$. Thus, $\mathbf{x}$ must be in $\mathcal{T}_3$, thus proving the lemma. □

Proof of Theorem 4.1. Let $\mathbf{y}$, k, and I be as in the statement of the theorem. Suppose that there exists a 2-coloring, $\chi : I \to \{1, 2\}$, such that whenever $i_1, i_2, i_3 \in I$ satisfy $\chi(i_1) = \chi(i_2) = \chi(i_3)$, then $i_1 + i_2 - i_3 \mod k \in I$. Define $\mathbf{x} = (x_j)_{j \in \mathbb{Z}} \in \{0,1,2\}^{\mathbb{Z}}$ as follows: $x_j = \chi(j \mod k)$ if $j \mod k \in I$, and $x_j = 0$ otherwise. It is easy to verify that $\mathbf{x} \in \mathcal{T}_3$, and $\operatorname{supp}(\mathbf{x}) = \bigcup_{i \in I}(k\mathbb{Z} + i) = \operatorname{supp}(\mathbf{y})$ so that $\mathbf{y} \in \mathcal{S}_3$.

If $\mathbf{y} \in \mathcal{S}_3$, then let $\mathbf{x}$, I_1, and I_2 be as in the statement of Lemma 4.5. Define the 2-coloring, χ, of I as follows: $\chi(j) = 1$ if $j \in I_1$, and $\chi(j) = 2$ if $j \in I_2$. From the fact that $\mathbf{x} \in \mathcal{T}_3$, it follows that χ has the property stated in the theorem. □

Our next goal is to provide a proof for Theorem 4.3. As is often the case, one direction of the theorem, namely the sufficiency of condition (i), (ii), or (iii), is easy to prove. Indeed, if $|I| \leq 2$, then any injective 2-coloring, χ, of I is a TGP-coloring. If $I = [0, p-1]$, then we may take $\chi(i) = 1$ for all $i \in I$ to be the required TGP-coloring. Finally, if $p = 5$, then one may actually verify by hand that each of the five 4-subsets of $\mathbb{Z}/5$ is in fact TGP-colorable.

We must now prove that one of conditions (i)–(iii) of Theorem 4.3 is necessary for the existence of a TGP-coloring of $I \subset \mathbb{Z}/p$. In fact, we need prove only that if $I \subset \mathbb{Z}/p$ is TGP-colorable and $|I| \geq 3$, then one of conditions (ii) and (iii) must hold. Note that $|I| \geq 3$ requires that $p \geq 3$, so we need not deal with $p = 2$.

Given a 2-coloring, χ, of $I \subset \mathbb{Z}/p$, we shall let $\{I_1, I_2\}$ denote the corresponding chromatic partition of I, i.e., for $j = 1, 2$, $I_j = \{i \in I : \chi(i) = j\}$. Observe that if $I \subset \mathbb{Z}/p$ is TGP-colorable and $|I| \geq 3$, then at least one of the following must be true:

(a) there exists a TGP-coloring, χ, of I such that $|I_1| \geq 2$ and $|I_2| \leq |I_1| - 2$;

(b) there exists a TGP-coloring, χ, of I such that $|I_1| \geq 2$ and $|I_2| = |I_1| - 1$;

(c) there exists a TGP-coloring, χ, of I such that $|I_1| \geq 2$ and $|I_2| = |I_1|$.

We shall analyze each of these cases separately and show that if (a) or (b) is true, then $I = \mathbb{Z}/p$, and if (c) is true, then condition (iii) of Theorem 4.3 holds.

The following lemma is the $\mathbb{Z}/p$-equivalent of Corollary 3.3 (for the case $q = 3$) and is the core ingredient in our proofs.

LEMMA 4.6. *Let χ be a TGP-coloring of $I \subset \mathbb{Z}/p$, for prime $p \geq 3$, and let $\{I_1, I_2\}$ be the corresponding chromatic partition of I. If either I_1 or I_2 contains a 3-term AP $\{a, a+d, a+2d\}$ for some $a, d \in \mathbb{Z}/p$, $d \neq 0$, then $I = \mathbb{Z}/p$.*

Proof. Let $\mathbf{x} = (x_k)_{k \in \mathbb{Z}} \in \{0,1,2\}^{\mathbb{Z}}$ be the periodic sequence defined as follows: $x_j = \chi(j \mod k)$ if $j \mod k \in I$, and $x_j = 0$ otherwise. From the conditions of the lemma, and recalling that the Schur number $S(2)$ equals 5, we see that $\mathbf{x}$ satisfies the hypotheses of Corollary 3.3 for $q = 3$. Hence, $a + d\mathbb{Z} \subset \text{supp}(\mathbf{x})$, from which it follows that I contains the set $K = \{a + jd \mod p : j \in \mathbb{Z}\}$. But, note that $H = \{jd \mod p : j \in \mathbb{Z}\}$ is a subgroup of $\mathbb{Z}/p$, and K is a coset of H. Since the only nonempty subgroup of $\mathbb{Z}/p$ is $\mathbb{Z}/p$ itself, it follows that $K = \mathbb{Z}/p$, which proves the lemma. □

The following proposition takes care of case (a) above.

PROPOSITION 4.7. *Let χ be a TGP-coloring of $I \subset \mathbb{Z}/p$, for prime $p \geq 3$, such that $|I_1| \geq 2$ and $|I_2| \leq |I_1| - 2$. Then, $I = \mathbb{Z}/p$.*

Proof. Let $I_1 = \{i_1, i_2, \ldots, i_m\}$, $m \geq 2$, so that $|I_2| \leq m - 2$. Let $j_k = 2i_1 - i_k$, $k = 2, 3, \ldots, m$. Since χ is a TGP-coloring of I, $j_k \in I = I_1 \cup I_2$ for $k = 2, 3, \ldots, m$. Now, note that the j_k's are all distinct since $j_k = j_l$ implies that $i_k = i_l$. Since there are $m - 1$ distinct j_k's and $|I_2| \leq m - 2$, there exists a $k \in [2, m]$ such that $j_k \in I_1$. But now, $\{i_k, i_1, j_k\}$ is a 3-term AP in I_1, and hence, by Lemma 4.6, $I = \mathbb{Z}/p$. □

To deal with the remaining cases (b) and (c), we need the following lemma.

LEMMA 4.8. *Let I_1, I_2 be disjoint subsets of $\mathbb{Z}/p$, for prime $p \geq 3$, with $|I_1| \geq 2$ and $|I_2| = |I_1| - 1$. If for all pairs of distinct $x_i, x_j \in I_1$, we have $2x_i - x_j \in I_2$, then $p = 3$ and $I_1 \cup I_2 = \mathbb{Z}/3$.*

Proof. Let $I_1 = \{x_1, x_2, \ldots, x_m\}$ for some $m \geq 2$ so that $|I_2| = m - 1$. Clearly, $m < p$, as otherwise I_1, I_2 cannot be disjoint. For each $i \in [1, m]$, define $Y_i = \{2x_i - x_j : j \in [1, m], j \neq i\}$. Similarly, for each $j \in [1, m]$, define $Z_j = \{2x_i - x_j : i \in [1, m], i \neq j\}$. By assumption, $Y_i, Z_j \subset I_2$ for all $i, j \in [1, m]$. Furthermore, note that for any fixed i, the elements of Y_i are all distinct in $\mathbb{Z}/p$. Therefore, for any $i \in [1, m]$, $|Y_i| = m - 1$, and hence, $Y_i = I_2$. Similarly, $Z_j = I_2$ for any $j \in [1, m]$.

Now, fix an arbitrary $a \in I_2$. For any $i \in [1, m]$, since $Y_i = I_2$, there exists a *unique* $j \in [1, m]$, $j \neq i$, such that $2x_i - x_j = a$. Therefore, we can define a function $\sigma : [1, m] \to [1, m]$ as follows: $\sigma(i)$ is the unique j such that $2x_i - x_j = a$. We shall show that σ is a bijection, which would imply that it is a permutation of $[1, m]$.

To show injectivity, we observe that if $\sigma(i) = \sigma(k) = j$ for some $i \neq k$, then we would have $2x_i - x_j = 2x_k - x_j$ for $i \neq k$. This would imply that $|Z_j| < m - 1$, which is impossible. To see that σ is surjective, take any $j \in [1, m]$, and note that $a \in Z_j$, as $Z_j = I_2$. Hence, there exists an $i \in [1, m]$ such that $2x_i - x_j = a$.

Thus, σ is a bijection, and hence, a permutation of $[1, m]$. Now, consider the m equations $2x_i - x_{\sigma(i)} = a$, $i = 1, 2, \dots, m$. Adding all these equations, we find

$$\begin{aligned} ma &= 2\sum_{i=1}^{m} x_i - \sum_{i=1}^{m} x_{\sigma(i)} \\ &= 2\sum_{i=1}^{m} x_i - \sum_{i=1}^{m} x_i \qquad (4) \\ &= \sum_{i=1}^{m} x_i \end{aligned}$$

with the equality in (4) following from the fact that σ is a permutation of $[1, m]$.

Since $2 \leq m < p$, there exists an $m^{-1} \in \mathbb{Z}/p$ such that $mm^{-1} = 1$. Therefore, $a = m^{-1}\sum_{i=1}^{m} x_i$. Since our choice of $a \in I_2$ was arbitrary, it follows that I_2 consists of the single element $a = m^{-1}\sum_{i=1}^{m} x_i$. Therefore, $m - 1 = |I_2| = 1$, which shows that $m = 2$.

We thus have $I_1 = \{x_1, x_2\}$ and $I_2 = \{a\}$, so that, by assumption, $2x_1 - x_2 = 2x_2 - x_1 = a$. But this means that $3(x_1 - x_2) = 0$ which, since $x_1 \neq x_2$, implies that $p = 3$. Therefore, since I_1 and I_2 are disjoint, we must have $I_1 \cup I_2 = \mathbb{Z}/p$. □

We can now readily dispose of case (b).

Proposition 4.9. *Let χ be a TGP-coloring of $I \subset \mathbb{Z}/p$, for prime $p \geq 3$, such that $|I_1| \geq 2$ and $|I_2| = |I_1| - 1$. Then, $I = \mathbb{Z}/p$.*

Proof. Since χ is a TGP-coloring of I, for any $x_i, x_j \in I_1$, $2x_i - x_j$ is either in I_1 or I_2. If for some pair of distinct $x_i, x_j \in I_1$, $2x_i - x_j$ is also in I_1, then $\{x_j, x_i, 2x_j - x_i\}$ is a 3-term AP in I_1, and hence $I = \mathbb{Z}/p$ by Lemma 4.6. If not, Lemma 4.8 applies, which also shows that $I = \mathbb{Z}/p$. □

We are now left with case (c) which, unfortunately, requires some work. We start with the following simple lemma.

Lemma 4.10. *Let χ be a TGP-coloring of $I \subset \mathbb{Z}/p$, for prime $p \geq 3$, such that $|I_1| = |I_2|$. Then, neither I_1 nor I_2 can contain a 3-term AP.*

Proof. If either I_1 or I_2 contains a 3-term AP, then by Lemma 4.6 $I = \mathbb{Z}/p$, implying that $|I| = p$, which is an odd number. However, $|I| = |I_1| + |I_2| = 2|I_1|$ is even. □

Thus, if x_1, x_2 is any pair of distinct elements in I_1, then $2x_2 - x_1 \in I_2$, for otherwise $\{x_1, x_2, 2x_2 - x_1\}$ would be a 3-term AP in I_1. By the same reasoning, $2y_2 - y_1 \in I_1$ for all $y_1, y_2 \in I_2$, $y_1 \neq y_2$.

The special case when $|I_1| = |I_2| = 2$ is straightforward, so we dispose of that first.

Lemma 4.11. *If χ is a TGP-coloring of $I \subset \mathbb{Z}/p$, for prime $p \geq 3$, such that $|I_1| = |I_2| = 2$, then $p = 5$.*

Proof. Let $I_1 = \{a, b\}$ for some $a, b \in [0, p-1]$, $a \neq b$. Since a, b are distinct elements in I_1, we must have $2a - b, 2b - a \in I_2$. Note that $2a - b \neq 2b - a$; otherwise, we would have $3(a - b) = 0$, implying that $p = 3$, which contradicts $p \geq |I_1| + |I_2| = 4$. Therefore, $I_2 = \{2a - b, 2b - a\}$.

Now, since $2a-b \neq 2b-a$, we must have $2(2a-b)-(2b-a) = 5a-4b \in I_1$. So, either $5a-4b = a$ or $5a-4b = b$. In the former case, we would get $4(a-b) = 0$, implying that $p|4$, which is impossible as $p \neq 2$. Therefore, we must have $5a-4b = b$, from which we obtain $5(a-b) = 0$, and hence $p = 5$ as desired. $\square$

The analysis of case (c) and, as a result, the proof of Theorem 4.3 would be complete if we can show that there cannot exist any TGP-coloring of $I \subset \mathbb{Z}/p$ such that $|I_1| = |I_2| \geq 3$. We show this in Proposition 4.14 below, but we need some development before we can prove the proposition.

Given a TGP-coloring, χ, of $I \subset \mathbb{Z}/p$ such that $|I_1| = |I_2| \geq 2$, we define certain functions $f, g : I_1 \to I_2$ as follows. Let $I_1 = \{x_1, x_2, \ldots, x_m\}$ and $I_2 = \{y_1, y_2, \ldots, y_m\}$, $m \geq 2$. For each $i \in [1,m]$, define the sets $Y_i = \{2x_j - x_i : \ j \in [1,m], j \neq i\}$ and $Z_i = \{2x_i - x_j : \ j \in [1,m], j \neq i\}$ so that $Y_i, Z_i \subset I_2$. For any fixed i, all the elements of Y_i are distinct, and hence $|Y_i| = m-1$. Since $|I_2| = m$, there is precisely one element in $I_2 \setminus Y_i$, and we shall denote this element by $f(x_i)$. Similarly, we denote the unique element in $I_2 \setminus Z_i$ by $g(x_i)$. Doing this for each $i \in [1,m]$, we get two mappings $f, g : I_1 \to I_2$. To be precise, $f(x_i) = y_j$ if and only if $I_2 \setminus Y_i = \{y_j\}$, and $g(x_i) = y_j$ if and only if $I_2 \setminus Z_i = \{y_j\}$. We make some observations about the sets Y_i, Z_i and the mappings f, g in the lemmas below.

Lemma 4.12. *For any $i \in [1,m]$, if $y, y' \in Y_i$ or $y, y' \in Z_i$, then $2y - y' \neq x_i$.*

Proof. We provide the argument only for $y, y' \in Y_i$, as the argument for $y, y' \in Z_i$ is similar. If $y, y' \in Y_i$, then there exist $x_k, x_l \in I_1$ such that $y = 2x_k - x_i$ and $y' = 2x_l - x_i$. So, if $2y - y' = x_i$, we would get $2(2x_k - x_l - x_i) = 0$, from which we obtain $2x_k - x_l = x_i$. But, this would mean that $\{x_l, x_k, x_i\}$ is a 3-term AP in I_1, contradicting Lemma 4.10. $\square$

Lemma 4.13. *f, g are bijections, and $f(x) = g(x)$ for all $x \in I_1$.*

Proof. We shall show that f is a bijection and that $f(x) = g(x)$ for all $x \in I_1$. It is then clear that g is also a bijection. Since I_1, I_2 are finite sets of the same cardinality, to prove that f is a bijection, it suffices to show that f is injective.

Now, suppose that f is not injective. Without loss of generality, assume that $f(x_{m-1}) = f(x_m) = y_m$. By the definition of f, $\{y_m\} = I_2 \setminus Y_{m-1} = I_2 \setminus Y_m$, and hence $Y_m = Y_{m-1} = \{y_1, y_2, \ldots, y_{m-1}\}$. Now, if $y_i, y_j \in Y_m = Y_{m-1}$ are such that $y_i \neq y_j$, then by Lemma 4.12, $2y_i - y_j \notin \{x_m, x_{m-1}\}$. However, as $2y_i - y_j$ must be in I_1, we find that $2y_i - y_j \in \{x_1, x_2, \ldots, x_{m-2}\}$. This means that the sets $\{y_1, y_2, \ldots, y_{m-1}\}$ and $\{x_1, x_2, \ldots, x_{m-2}\}$ satisfy the assumptions of Lemma 4.8, and therefore, we must have $p = 3$. But this is impossible as $p \geq |I_1| + |I_2| = 2m \geq 4$. This shows that f is injective and hence a bijection.

To show that $f(x) = g(x)$ for all $x \in I_1$, suppose to the contrary that $g(x_m) = y_m$, but $f(x_m) \neq y_m$. Now, since f is a bijection, there exists an $x \in I$, $x \neq x_m$, such that $f(x) = y_m$. Relabeling the x_i's if necessary, we may take $f(x_{m-1}) = y_m$. We thus have $f(x_{m-1}) = g(x_m) = y_m$. Now, using the same argument as used earlier to prove the injectivity of f, except that now we replace Y_m by Z_m, we reach the conclusion via Lemma 4.8 that $p = 3$, which is impossible. Thus, we must have $f(x) = g(x)$ for all $x \in I_1$. $\square$

We are now ready to prove the following result, which is the last step in our proof of Theorem 4.3.

Proposition 4.14. *For any $I \subset \mathbb{Z}/p$, $p \geq 3$, there does not exist a TGP-coloring of I such that $|I_1| = |I_2| \geq 3$.*

Proof. Suppose there exists such a coloring of I. Let $I_1 = \{x_1, x_2 \ldots, x_m\}$ and $I_2 = \{y_1, y_2, \ldots, y_m\}$, $m \geq 3$. Note that $\{2y_i - y_1 : \ i \in [2,m]\}$ lies in I_1, and all its

$m-1$ elements are distinct. By relabeling the x_j's if necessary, we may assume that $2y_i - y_1 = x_i$ for all $i \in [2,m]$.

Let $f, g : I_1 \to I_2$ be the mappings defined as above. We shall show that for any $i \in [2,m]$, $f(x_i) = y_1$. But this leads to a contradiction since for $m \geq 3$, there exist $x_2, x_3 \in I_1$, $x_2 \neq x_3$, for which $f(x_2) = f(x_3) = y_1$, which is impossible, as f is a bijection.

So, consider an arbitrary $x_i \in I_1$ with $i \in [2,m]$, and suppose that $f(x_i) \neq y_1$. Thus,

$$x_i = 2y_i - y_1, \tag{5}$$

and there exists an $x_j \in I_1$ such that

$$2x_j - x_i = y_1. \tag{6}$$

If $j \geq 2$ as well, then we would also have

$$x_j = 2y_j - y_1. \tag{7}$$

Therefore, plugging (5) and (7) into (6), we would obtain $2(2y_j - y_i - y_1) = 0$, implying that $2y_j - y_i = y_1$, which is impossible by Lemma 4.10. Thus, $j = 1$, so we must have

$$2x_1 - x_i = y_1. \tag{8}$$

Since, by Lemma 4.13, $f(x_i) = g(x_i)$, we also have $g(x_i) \neq y_1$. Now, an argument similar to the one above shows that

$$2x_i - x_1 = y_1. \tag{9}$$

But from (8) and (9), we get $2x_1 - x_i = 2x_i - x_1$ or, equivalently, $3(x_i - x_1) = 0$ which is impossible, as $p \geq |I_1| + |I_2| = 2m \geq 6$.

Thus, we are forced to conclude that $f(x_i) = y_1$, and since this holds for any $i \in [2,m]$, this contradicts the fact that f is a bijection. □

This concludes our proof of Theorem 4.3.

4.2. Aperiodic sequences in $\mathcal{S}_3$. We shall now work towards a proof for Theorem 4.4. It is easy to show the sufficiency of condition (i), (ii), or (iii) in the statement of the theorem, so we proceed to do that first. For $\mathbf{y} \in \{0,1\}^{\mathbb{Z}}$ such that condition (i) holds, we construct an $\mathbf{x} = (x_j)_{j\in\mathbb{Z}} \in \mathcal{T}_3$ with $\operatorname{supp}(\mathbf{x}) = \operatorname{supp}(\mathbf{y})$ as follows. If $\operatorname{supp}(\mathbf{y}) = \{m\}$ for some $m \in \mathbb{Z}$, then simply take $\mathbf{x} = \mathbf{y}$; if $\operatorname{supp}(\mathbf{y}) = \{m,n\}$ for $m,n \in \mathbb{Z}$, $m \neq n$, then set $x_m = 1$, $x_n = 2$, and $x_j = 0$ otherwise. For $\mathbf{y} \in \{0,1\}^{\mathbb{Z}}$ such that $\operatorname{supp}(\mathbf{y}) = (k\mathbb{Z}+i) \cup V$ as in condition (ii), let $\mathbf{x} \in \{0,1,2\}^{\mathbb{Z}}$ be the sequence for which $V_1(\mathbf{x}) = k\mathbb{Z}+i$ and $V_2(\mathbf{x}) = V$. For $\mathbf{y} \in \{0,1\}^{\mathbb{Z}}$ such that $\operatorname{supp}(\mathbf{y}) = (3t\mathbb{Z}+i) \cup V$ as in condition (iii), let $\mathbf{x} \in \{0,1,2\}^{\mathbb{Z}}$ be the sequence for which $V_1(\mathbf{x}) = 3t\mathbb{Z}+i$ and $V_2(\mathbf{x}) = V$. In both of these cases, it is straightforward to verify that $\mathbf{x} \in \mathcal{T}_3$, and hence, $\mathbf{y} = \pi(\mathbf{x}) \in \mathcal{S}_3$.

To prove the converse part of the theorem, we use the following approach. We first show that if $\mathbf{y} \in \mathcal{S}_3$ is such that $\overline{d}(\operatorname{supp}(\mathbf{y})) = 0$, then $|\operatorname{supp}(\mathbf{y})| \leq 2$. Thus, if $\mathbf{y} \in \mathcal{S}_3$ is aperiodic with $\overline{d}(\operatorname{supp}(\mathbf{y})) = 0$, then we must have $1 \leq |\operatorname{supp}(\mathbf{y})| \leq 2$ since $|\operatorname{supp}(\mathbf{y})| = 0$ implies that $\mathbf{y}$ is the all-zeros sequence, which is periodic. For aperiodic $\mathbf{y} \in \mathcal{S}_3$ with $\overline{d}(\operatorname{supp}(\mathbf{y})) > 0$, we analyze sequences in the set $\pi^{-1}(\mathbf{y}) \cap \mathcal{T}_3$, finally showing that there must exist a sequence $\mathbf{x} \in \pi^{-1}(\mathbf{y}) \cap \mathcal{T}_3$ such that $V_1(\mathbf{x}) = k\mathbb{Z} + i$, for some $k \in \mathbb{N}$ and $i \in [0, k-1]$, and $V_2(\mathbf{x})$ is one of the V's in conditions (ii) and (iii) in the statement of Theorem 4.4.

LEMMA 4.15. *If* $\mathbf{y} \in \mathcal{S}_3$ *is such that* $\overline{d}(\operatorname{supp}(\mathbf{y})) = 0$*, then* $|\operatorname{supp}(\mathbf{y})| \leq 2$*.*

Proof. Suppose that $\mathbf{y} \in \mathcal{S}_3$ is such that $|\operatorname{supp}(\mathbf{y})| \geq 3$. We shall show that $\overline{d}(\operatorname{supp}(\mathbf{y})) > 0$. Let $\mathbf{x} \in \mathcal{T}_3$ be any sequence with $\pi(\mathbf{x}) = \mathbf{y}$ so that $|V_1(\mathbf{x})| + |V_2(\mathbf{x})| = |\operatorname{supp}(\mathbf{x})| \geq 3$. Thus, either $|V_1(\mathbf{x})| \geq 2$ or $|V_2(\mathbf{x})| \geq 2$. We shall assume that $|V_1(\mathbf{x})| \geq 2$, as a symmetric argument applies to the other case. Our goal is to show that either $V_1(\mathbf{x})$ or $V_2(\mathbf{x})$ contains a 3-term AP $\{a, a+k, a+2k\}$ for some $a \in \mathbb{Z}$ and $k \in \mathbb{N}$. For if this is the case, then applying Corollary 3.3 with $q = 3$, noting that $S(2) = 5$, we find that $a + k\mathbb{Z} \subset \operatorname{supp}(\mathbf{x})$. Therefore, we would have $\overline{d}(\operatorname{supp}(\mathbf{y})) = \overline{d}(\operatorname{supp}(\mathbf{x})) \geq \overline{d}(a + k\mathbb{Z}) = 1/k > 0$, which proves the lemma.

Since $|V_1(\mathbf{x})| \geq 2$, pick any pair of integers $r, s \in V_1(\mathbf{x})$, $r < s$, and let $d = s - r$. Now, suppose that neither $V_1(\mathbf{x})$ nor $V_2(\mathbf{x})$ contains a 3-term AP. Since $\mathbf{x} \in \mathcal{T}_3$ and $r, r+d \in V_1(\mathbf{x})$, we must have $r + 2d \in V_1(\mathbf{x}) \cup V_2(\mathbf{x})$ as $r + 2d = (r+d) + (r+d) - r$. But as $V_1(\mathbf{x})$ does not contain a 3-term AP, $r + 2d \in V_2(\mathbf{x})$. Similarly, $r - d \in V_2(\mathbf{x})$ as $r - d = r + r - (r+d)$. Now, applying a similar argument to the pair of integers $r - d, r + 2d \in V_2(\mathbf{x})$, we find that $r - 4d, r + 5d \in V_1(\mathbf{x})$. Next, $r + d, r + 5d \in V_1(\mathbf{x})$ implies that $r - 3d \in V_2(\mathbf{x})$. Finally, since $r - d, r - 3d \in V_2(\mathbf{x})$, we have $r - 5d \in V_1(\mathbf{x})$. But now, $\{r - 5d, r, r + 5d\}$ is a 3-term AP in $V_1(\mathbf{x})$, contradicting our assumption. Hence, if $|V_1(\mathbf{x})| \geq 2$, then either $V_1(\mathbf{x})$ or $V_2(\mathbf{x})$ contains a 3-term AP, thus proving the lemma. □

As explained earlier, the above lemma shows that if $\mathbf{y} \in \mathcal{S}_3$ is aperiodic with $\overline{d}(\operatorname{supp}(\mathbf{y})) = 0$, then $1 \leq \operatorname{supp}(\mathbf{y}) \leq 2$, which is condition (i) of Theorem 4.4.

So now, we are left to deal with the set of aperiodic sequences $\mathbf{y} \in \mathcal{S}_3$ with $\overline{d}(\operatorname{supp}(\mathbf{y})) > 0$, which we shall denote by $\mathcal{Q}_3$. As before, for $\mathbf{x} \in \mathcal{T}_3$, we define $P(\mathbf{x}) = \bigcup_{j \in J_1} V_j(\mathbf{x})$, where $J_1 = \{j \in \{1,2\} : \overline{d}(V_j(\mathbf{x})) > 0\}$, and $N(\mathbf{x}) = \bigcup_{j \in J_2} V_j(\mathbf{x})$, where $J_2 = \{j \in \{1,2\} : \overline{d}(V_j(\mathbf{x})) = 0\}$. Note that if $\mathbf{x} \in \mathcal{T}_3$ is such that $\pi(\mathbf{x}) \in \mathcal{Q}_3$, then $P(\mathbf{x}) \neq \emptyset$ since $\overline{d}(\operatorname{supp}(\mathbf{x})) = \overline{d}(\operatorname{supp}(\pi(\mathbf{x}))) > 0$, and by Corollary 3.5, $N(\mathbf{x}) \neq \emptyset$ as well since $\pi(\mathbf{x})$ is aperiodic. Thus, for any $\mathbf{x} \in \mathcal{T}_3$ such that $\pi(\mathbf{x}) \in \mathcal{Q}_3$, we have $\{P(\mathbf{x}), N(\mathbf{x})\} = \{V_1(\mathbf{x}), V_2(\mathbf{x})\}$.

The proof of the remaining part of Theorem 4.4 begins with the following lemma.

LEMMA 4.16. *Let* $\mathbf{x} \in \mathcal{T}_3$ *be such that* $\pi(\mathbf{x}) \in \mathcal{Q}_3$*. If there exists a* $k \in \mathbb{N}$ *and an* $I \subset [0, k-1]$ *such that* $P(\mathbf{x}) \subset \bigcup_{i \in I}(k\mathbb{Z} + i) \subset \operatorname{supp}(\mathbf{x})$*, then the elements of* $\operatorname{supp}(\mathbf{x}) \setminus \bigcup_{i \in I}(k\mathbb{Z}+i)$ *are all distinct modulo* k*. Hence,* $|\operatorname{supp}(\mathbf{x}) \setminus \bigcup_{i \in I}(k\mathbb{Z}+i)| \leq k$*.*

Proof. Let $N'(\mathbf{x}) = \operatorname{supp}(\mathbf{x}) \setminus \bigcup_{i \in I}(k\mathbb{Z} + i)$. Note that under the assumptions of the lemma, $N'(\mathbf{x}) \subset \operatorname{supp}(\mathbf{x}) \setminus P(\mathbf{x}) = N(\mathbf{x})$. We shall show that if $N'(\mathbf{x})$ contains distinct integers a, b such that $a \equiv b \pmod{k}$, then $\overline{d}(N'(\mathbf{x})) > 0$. This leads to the contradiction that $\overline{d}(N(\mathbf{x})) > 0$ since $\overline{d}(N(\mathbf{x})) \geq \overline{d}(N'(\mathbf{x}))$.

So, suppose that $a, b \in N'(\mathbf{x})$, $a < b$, are such that $a \equiv b \pmod{k}$. For any $j \in \operatorname{supp}(\mathbf{x})$, by the definition of $N'(\mathbf{x})$, $j \in N'(\mathbf{x})$ if and only if $j \mod k \notin I$. In particular, $a \equiv b \equiv \ell \pmod{k}$ for some $\ell \notin I$. Let $d = b - a$, and note that $d \equiv 0 \pmod{k}$.

As observed above, for any $\mathbf{x} \in \mathcal{T}_3$ such that $\pi(\mathbf{x}) \in \mathcal{Q}_3$, we have $\{P(\mathbf{x}), N(\mathbf{x})\} = \{V_1(\mathbf{x}), V_2(\mathbf{x})\}$. Without loss of generality, we may assume that $P(\mathbf{x}) = V_1(\mathbf{x})$ and $N(\mathbf{x}) = V_2(\mathbf{x})$, and hence, $N'(\mathbf{x}) \subset V_2(\mathbf{x})$. So, we have $a, a+d \in V_2(\mathbf{x})$, and hence by the TGP condition, $a + 2d \in \operatorname{supp}(\mathbf{x})$. But since $a + 2d \equiv \ell \pmod{k}$ and $\ell \notin I$, we must have $a + 2d \in N'(\mathbf{x}) \subset V_2(\mathbf{x})$. But now, $V_2(\mathbf{x})$ contains the 3-term AP $\{a, a+d, a+2d\}$ so that by Corollary 3.3, $a + d\mathbb{Z} \subset \operatorname{supp}(\mathbf{x})$. However, for any $j \in a + d\mathbb{Z}$, $j \equiv \ell \pmod{k}$, and hence $j \in N'(\mathbf{x})$. Thus, $a + d\mathbb{Z} \subset N'(\mathbf{x})$, from which we obtain $\overline{d}(N'(\mathbf{x})) \geq \overline{d}(a + d\mathbb{Z}) = 1/d > 0$, which leads to the contradiction that proves the lemma. $\square$

Lemma 4.16 leads us to the following result, which is the crucial step in our proof of the rest of the converse part of Theorem 4.4.

LEMMA 4.17. *For each $\mathbf{y} \in \mathcal{Q}_3$, there exists an $\mathbf{x} \in \mathcal{T}_3$ such that $\pi(\mathbf{x}) = \mathbf{y}$ and $P(\mathbf{x}) = \bigcup_{i\in I}(k\mathbb{Z} + i)$ for some $k \in \mathbb{N}$ and $I \subset [0, k-1]$.*

Proof. Consider an arbitrary $\mathbf{y} \in \mathcal{Q}_3$. Since $\mathcal{Q}_3 \subset \mathcal{S}_3$, there exists a $\mathbf{z} \in \mathcal{T}_3$ such that $\pi(\mathbf{z}) = \mathbf{y}$. As observed prior to Lemma 4.16, for such a $\mathbf{z}$, we have $\{P(\mathbf{z}), N(\mathbf{z})\} = \{V_1(\mathbf{z}), V_2(\mathbf{z})\}$. Without loss of generality, we may assume that $P(\mathbf{z}) = V_1(\mathbf{z})$ and $N(\mathbf{z}) = V_2(\mathbf{z})$. Since $\overline{d}(V_1(\mathbf{z})) = \overline{d}(P(\mathbf{z})) > 0$, by Lemma 3.4, there exist a $k \in \mathbb{N}$ and an $I \subset [0, k-1]$ such that $V_1(\mathbf{z}) \subset \bigcup_{i\in I}(k\mathbb{Z}+i) \subset \operatorname{supp}(\mathbf{z})$. Moreover, by Lemma 4.16, $\operatorname{supp}(\mathbf{z}) \setminus \bigcup_{i\in I}(k\mathbb{Z} + i)$ is a finite set.

Note that for any $i \in I$, $k\mathbb{Z} + i \subset \operatorname{supp}(\mathbf{z}) = V_1(\mathbf{z}) \cup V_2(\mathbf{z})$ so that $\overline{d}(k\mathbb{Z} + i) = \overline{d}(V_1(\mathbf{z}) \cap (k\mathbb{Z} + i)) + \overline{d}(V_2(\mathbf{z}) \cap (k\mathbb{Z} + i))$. Since $\overline{d}(V_2(\mathbf{z}) \cap (k\mathbb{Z} + i)) = 0$, we have $\overline{d}(V_1(\mathbf{z}) \cap (k\mathbb{Z}+i)) = \overline{d}(k\mathbb{Z}+i) = 1/k > 0$. In particular, $V_1(\mathbf{z}) \cap (k\mathbb{Z}+i)$ is an infinite set for any $i \in I$.

Now, let $\mathbf{x} = (x_j)_{j\in\mathbb{Z}} \in \{0, 1, 2\}^{\mathbb{Z}}$ be defined as follows:

$$x_j = \begin{cases} 0 & \forall\ j \notin \operatorname{supp}(\mathbf{z}), \\ 1 & \forall\ j \in \bigcup_{i\in I}(k\mathbb{Z} + i), \\ 2 & \forall\ j \in \operatorname{supp}(\mathbf{z}) \setminus \bigcup_{i\in I}(k\mathbb{Z} + i). \end{cases}$$

It is clear that $\pi(\mathbf{x}) = \mathbf{y}$ since $\operatorname{supp}(\mathbf{x}) = \operatorname{supp}(\mathbf{z}) = \operatorname{supp}(\mathbf{y})$. Also, we have $\overline{d}(V_1(\mathbf{x})) \geq 1/k > 0$, and since $V_2(\mathbf{x})$ is a finite set, $\overline{d}(V_2(\mathbf{x})) = 0$. Hence, $P(\mathbf{x}) = V_1(\mathbf{x}) = \bigcup_{i\in I}(k\mathbb{Z} + i)$. It remains only to show that $\mathbf{x} \in \mathcal{T}_3$, i.e., to show that if $p, q, r \in V_1(\mathbf{x})$ or $p, q, r \in V_2(\mathbf{x})$, then $p + q - r \in \operatorname{supp}(\mathbf{x})$.

Note that $V_2(\mathbf{x}) \subset V_2(\mathbf{z})$. Therefore, if we take any $p, q, r \in V_2(\mathbf{x})$, then $p, q, r \in V_2(\mathbf{z})$ as well, and hence, since $\mathbf{z} \in \mathcal{T}_3$, $p + q - r \in \operatorname{supp}(\mathbf{z}) = \operatorname{supp}(\mathbf{x})$.

Now, let $p, q, r \in V_1(\mathbf{x})$, and suppose that $p+q-r \notin \operatorname{supp}(\mathbf{x})$. Thus, $p+q-r \equiv j \pmod{k}$ for some $j \notin I$. As shown above, for any $i \in I$, $V_1(\mathbf{z}) \cap (k\mathbb{Z} + i)$ is an infinite set. Hence, we can pick $q', r' \in V_1(\mathbf{z})$ such that $q' \equiv q \pmod{k}$ and $r' \equiv r \pmod{k}$, and furthermore, we can pick a $p' \in V_1(\mathbf{z})$, with $p' \equiv p \pmod{k}$, that is large enough in absolute value that $p' + q' - r'$ lies outside the finite set $\operatorname{supp}(\mathbf{z}) \setminus \bigcup_{i\in I}(k\mathbb{Z} + i)$. Thus, $p'+q'-r' \notin \operatorname{supp}(\mathbf{z}) \setminus \bigcup_{i\in I}(k\mathbb{Z}+i)$, and as $p'+q'-r' \equiv p+q-r \equiv j \pmod{k}$, we see that $p' + q' - r' \notin \bigcup_{i\in I}(k\mathbb{Z} + i)$ either. This shows that $p' + q' - r' \notin \operatorname{supp}(\mathbf{z})$, which contradicts the fact that $\mathbf{z} \in \mathcal{T}_3$. Hence, we must have $p + q - r \in \operatorname{supp}(\mathbf{x})$, which shows that $\mathbf{x} \in \mathcal{T}_3$, thus proving the result. $\square$

In the next two lemmas, we show that the sequence $\mathbf{x} \in \mathcal{T}_3$, whose existence is guaranteed by Lemma 4.17, must in fact have $P(\mathbf{x}) = k_0\mathbb{Z} + i_0$, for some $k_0 \in \mathbb{N}$ and $i_0 \in [0, k_0 - 1]$, and $N(\mathbf{x}) = V$, where V is as in condition (ii) or (iii) of the theorem.

LEMMA 4.18. *Let $\mathbf{x} \in \mathcal{T}_3$ be such that $\pi(\mathbf{x}) \in \mathcal{Q}_3$. If there exist a $k \in \mathbb{N}$ and an $I \subset [0, k-1]$ such that $P(\mathbf{x}) = \bigcup_{i\in I}(k\mathbb{Z} + i)$, then $\bigcup_{i\in I}(k\mathbb{Z} + i) = d\mathbb{Z} + \ell$ for some $d \in \mathbb{N}$, such that $d|k$, and some $\ell \in [0, d-1]$.*

Proof. Our goal is to show that under the assumptions of the lemma, I must be a coset of some subgroup of the group, $\mathbb{Z}/k$, of integers modulo k. (We represent $\mathbb{Z}/k$ here as the set $[0, k-1]$ equipped with the operation of modulo-k addition.) Since any (nonempty) subgroup of $\mathbb{Z}/k$ is generated by some divisor d of k, any such I must be of the form $\{i \in [0, k-1] :\ i \equiv \ell \pmod{d}\}$ for some $d \in \mathbb{N}$, $d|k$, and some $\ell \in [0, d-1]$. It then immediately follows that $\bigcup_{i\in I}(k\mathbb{Z} + i) = d\mathbb{Z} + \ell$, as stated in the lemma.

Now, to show that I is a coset of some subgroup of $\mathbb{Z}/k$, it is enough to show that I is closed under the ternary operation $i_1 + i_2 - i_3 \mod k$; i.e., if $i_1, i_2, i_3 \in I$, then $i_1 + i_2 - i_3 \mod k \in I$. Indeed, if I is closed under this operation, then take any $\ell \in I$ and consider the set $H = \{i - \ell \mod k : i \in I\}$. It is easily verified that H is a subgroup of $\mathbb{Z}/k$, and so I is a coset of H.

Thus, it remains only to prove that under the assumptions of the lemma, if $i_1, i_2, i_3 \in I$, then $i_1 + i_2 - i_3 \mod k \in I$. As $\mathbf{x} \in \mathcal{T}_3$ is such that $\pi(\mathbf{x}) \in \mathcal{Q}_3$, we may assume that $P(\mathbf{x}) = V_1(\mathbf{x})$ and $N(\mathbf{x}) = V_2(\mathbf{x})$. Thus, $V_1(\mathbf{x}) = \bigcup_{i \in I}(k\mathbb{Z} + i)$ and $V_2(\mathbf{x}) = \text{supp}(\mathbf{x}) \setminus \bigcup_{i \in I}(k\mathbb{Z} + i)$. Note that k and I satisfy the assumptions of Lemma 4.16, and hence, we have $|V_2(\mathbf{x})| = |\text{supp}(\mathbf{x}) \setminus \bigcup_{i \in I}(k\mathbb{Z} + i)| \leq k$. In other words, $V_2(\mathbf{x})$ is a finite set.

Now, consider any $i_1, i_2, i_3 \in I$. Since $V_2(\mathbf{x})$ is a finite set, we can choose an integer r large enough that $kr + (i_1 + i_2 - i_3) \notin V_2(\mathbf{x})$. Let $r_1, r_2, r_3 \in \mathbb{Z}$ be such that $r_1 + r_2 - r_3 = r$. Note that for $j = 1, 2, 3$, $kr_j + i_j \in V_1(\mathbf{x})$ since $k\mathbb{Z} + i_j \subset V_1(\mathbf{x})$. Hence, by the TGP condition applied to $kr_1 + i_1, kr_2 + i_2, kr_3 + i_3 \in V_1(\mathbf{x})$, we obtain $kr + (i_1 + i_2 - i_3) \in \text{supp}(\mathbf{x})$. Since $kr + (i_1 + i_2 - i_3)$ is not in $V_2(\mathbf{x})$, it must be in $V_1(\mathbf{x}) = \bigcup_{i \in I}(k\mathbb{Z} + i)$, from which it follows that $i_1 + i_2 - i_3 \mod k \in I$, as desired. □

LEMMA 4.19. *Let $\mathbf{x} \in \mathcal{T}_3$ be such that $\pi(\mathbf{x}) \in \mathcal{Q}_3$ and $P(\mathbf{x}) = k\mathbb{Z} + i$ for some $k \in \mathbb{N}$ and $i \in [0, k-1]$. Then, $1 \leq |N(\mathbf{x})| \leq 2$. Furthermore, if $|N(\mathbf{x})| = 2$, then $k = 3t$ for some $t \in \mathbb{N}$ and $N(\mathbf{x}) \equiv \{t + i, 2t + i\} \pmod{3t}$.*

Proof. Since $\mathbf{x}$ satisfies the assumptions of Lemma 4.16 and $N(\mathbf{x}) = \text{supp}(\mathbf{x}) \setminus (k\mathbb{Z} + i)$, we find that $|N(\mathbf{x})| \leq k$. Furthermore, if a, b are distinct integers in $N(\mathbf{x})$, then $a \not\equiv b \pmod{k}$. As usual, we shall assume that $P(\mathbf{x}) = V_1(\mathbf{x})$ and $N(\mathbf{x}) = V_2(\mathbf{x})$ so that $V_1(\mathbf{x}) = k\mathbb{Z} + i$ and $V_2(\mathbf{x})$ is a finite set. Furthermore, since $\pi(\mathbf{x})$ is aperiodic, $V_2(\mathbf{x})$ cannot be empty, i.e., $|V_2(\mathbf{x})| \geq 1$. We shall show that if $|V_2(\mathbf{x})| > 1$, then we must have $|V_2(\mathbf{x})| = 2$, $k = 3t$ for some $t \in \mathbb{N}$, and $V_2(\mathbf{x}) \equiv \{t + i, 2t + i\} \pmod{3t}$, which would prove the lemma.

So, suppose that $|V_2(\mathbf{x})| \geq 2$. Let a be the smallest integer in $V_2(\mathbf{x})$ and let b be the largest, so that $a < b$. Note that since a and b are distinct integers in $V_2(\mathbf{x}) = N(\mathbf{x})$, we have $a \not\equiv b \pmod{k}$. Now, applying the TGP condition to $a, b \in V_2(\mathbf{x})$, we find that $2a - b, 2b - a \in \text{supp}(\mathbf{x})$. However, $2a - b < a$, and since a is the smallest integer in $V_2(\mathbf{x})$, $2a - b$ cannot be in $V_2(\mathbf{x})$. Hence, $2a - b \in V_1(\mathbf{x})$. A similar argument shows that $2b - a \in V_1(\mathbf{x})$ as well. But since $V_1(\mathbf{x}) = k\mathbb{Z} + i$, we have

$$2a - b \equiv 2b - a \equiv i \pmod{k}. \tag{10}$$

Therefore, $3(a - b) \equiv 0 \pmod{k}$. Since $a \not\equiv b \pmod{k}$, 3 must divide k and $a \equiv b \pmod{k/3}$.

Thus, $k = 3t$ for some $t \in \mathbb{N}$, and so we have $a \equiv b \pmod{t}$, but $a \not\equiv b \pmod{3t}$. Therefore, either $b \equiv t + a \pmod{3t}$ or $b \equiv 2t + a \pmod{3t}$. But since a, b must also satisfy the congruence $2b - a \equiv i \pmod{3t}$ in (10), some simple manipulations now show that $\{a, b\} \equiv \{t + i, 2t + i\} \pmod{3t}$.

Now, if there exists a $c \in V_2(\mathbf{x})$ such that $a < c < b$, then a similar argument as that used for (10) establishes that $a + c - b \equiv b + c - a \equiv i \pmod{k}$, and hence

$$2(a - b) \equiv 0 \pmod{k}. \tag{11}$$

Using $k = 3t$ and $\{a, b\} \equiv \{t + i, 2t + i\} \pmod{3t}$, it follows from (11) that either $t \equiv 0 \pmod{3t}$ or $2t \equiv 0 \pmod{3t}$, both of which are impossible for $t \neq 0$. Hence, if

TABLE 5.1
The number, $P(k)$, of subsets of $\mathbb{Z}/k$ that are TGP-colorable for nonprime $k \leq 20$.

k	1	4	6	8	9	10	12	14	15	16	18	20
$P(k)$	2	16	52	80	98	134	340	228	328	384	808	746

$|V_2(\mathbf{x})| > 1$, then $V_2(\mathbf{x})$ cannot contain anything other than the two integers a, b as above, which completes the proof of the lemma. $\square$

From Lemmas 4.17, 4.18, and 4.19, we see that for any $\mathbf{y} \in \mathcal{Q}_3$, either $\text{supp}(\mathbf{y})$ is of the form given in condition (ii), or it must be as in condition (iii) of Theorem 4.4, which completes the proof of that theorem.

5. Numerical results and conjectures. For $k \in \mathbb{N}$, let $P(k)$ denote the number of TGP-colorable subsets of $\mathbb{Z}/k$. It follows from Theorem 4.3 that $P(2) = 4$, $P(3) = 8$, $P(5) = 22$, and for primes $p > 5$, $P(p) = 1 + p + \binom{p}{2} + 1 = p(p+1)/2 + 2$. However, for nonprime k, we do not have a simple means of computing $P(k)$ as we do not have a complete solution to the problem of determining precisely which subsets of $\mathbb{Z}/k$ are TGP-colorable. We list the values of $P(k)$ for nonprime $k \leq 20$ in Table 5.1 below, most of which have been obtained by means of an exhaustive computer search.

Table 5.1 seems to suggest that $P(k)$ grows slowly, perhaps polynomially, with k. Now, recall our definition of $\mathcal{B}_{3,n}$ as the set of all n-blocks of $\mathcal{S}_3$. It can be inferred from Theorem 4.4 that aperiodic sequences in $\mathcal{S}_3$ contribute $O(n^4)$ blocks to $\mathcal{B}_{3,n}$. Based on the slow growth rate of $P(k)$, we conjecture that the number of blocks contributed to $\mathcal{B}_{3,n}$ by periodic sequences in $\mathcal{S}_3$ is also polynomial in n. Consequently, we conjecture that $h(\mathcal{S}_3) = 0$, just as in the BGP case.

REFERENCES

[1] M. J. ABLOWITZ AND T. HIROOKA, *Intrachannel pulse interactions in dispersion-managed transmission systems: Energy transfer*, Opt. Lett., 27 (2002), pp. 203–205.

[2] N. ALIC AND Y. FAINMAN, *Data dependent phase coding for mitigation of intrachannel four wave mixing*, in Proceedings of the 15th Annual Meeting of the IEEE Lasers and Electro-Optics Society (LEOS), 2002, pp. 699–700.

[3] R. L. GRAHAM, *Rudiments of Ramsey Theory*, Regional Conference Series in Mathematics 45, American Mathematical Society, Providence, RI, 1981.

[4] R. L. GRAHAM, B. L. ROTHSCHILD, AND J. H. SPENCER, *Ramsey Theory*, Wiley-Interscience, New York, 1980.

[5] N. KASHYAP, P. H. SIEGEL, AND A. VARDY, *Coding for the optical channel—the ghost pulse constraint*, IEEE Trans. Inform. Theory, submitted.

[6] D. LIND AND B. MARCUS, *An Introduction to Symbolic Dynamics and Coding*, Cambridge University Press, Cambridge, UK, 1995.

[7] X. LIU, X. WEI, A. H. GNAUCK, C. XU, AND L. K. WICKHAM, *Suppression of intrachannel four-wave-mixing induced ghost pulses in high-speed transmissions by phase inversion between adjacent marker blocks*, Opt. Lett., 27 (2002), pp. 1177–1179.

[8] N. J. A. SLOANE, *The Online Encyclopedia of Integer Sequences*, http://www.research.att.com/~njas/sequences/.

[9] J. ZWECK AND C. R. MENYUK, *Analysis of four-wave mixing between pulses in high-data-rate quasi-linear subchannel-multiplexed systems*, Opt. Lett., 27 (2002), pp. 1235–1237.

SIAM J. Discrete Math.
Vol. 19, No. 4, pp. 938–951

DYNAMIC TCP ACKNOWLEDGMENT: PENALIZING LONG DELAYS[*]

SUSANNE ALBERS[†] AND HELGE BALS[‡]

Abstract. We study the problem of acknowledging a sequence of data packets that are sent across a TCP connection. Previous work on the problem has focused mostly on the objective function that minimizes the sum of the number of acknowledgments sent and on the delays incurred for all of the packets. Dooly, Goldman, and Scott presented a deterministic 2-competitive online algorithm and showed that this is the best competitiveness of a deterministic strategy. Recently Karlin, Kenyon, and Randall developed a randomized online algorithm that achieves an optimal competitive ratio of $e/(e-1) \approx 1.58$.

In this paper we investigate a new objective function that minimizes the sum of the number of acknowledgments sent and the *maximum delay* incurred for any of the packets. This function is especially interesting if a TCP connection is used for interactive data transfer between network nodes. The TCP acknowledgment problem with this new objective function is different in structure than the problem with the function considered previously. We develop a deterministic online algorithm that achieves a competitive ratio of $\pi^2/6 \approx 1.644$ and prove that no deterministic algorithm can have a smaller competitiveness. We also study a generalized objective function where delays are taken to the pth power for some positive integer p. Again we give tight upper and lower bounds on the best possible competitive ratio of deterministic online algorithms. The competitiveness is 1 plus an alternating sum of Riemann's zeta function and tends to 1.5 as $p \to \infty$. Finally, we consider randomized online algorithms and show that, for our first objective function, no randomized strategy can achieve a competitive ratio smaller than $3/(3-2/e) \approx 1.324$. For the generalized objective function we show a lower bound of $2/(2-1/e) \approx 1.225$.

Key words. online algorithm, competitive analysis, TCP, data packet, acknowledgment

AMS subject classifications. 68Q25, 68W25, 68W40

DOI. 10.1137/S0895480104441656

1. Introduction.

Dooly, Goldman, and Scott [2, 3] recently initiated the algorithmic study of the *dynamic TCP acknowledgment problem*. In large networks such as the Internet, data transmission is performed using the TCP. Consider an open TCP connection between two network nodes that wish to exchange data. The data is partitioned into segments or *packets* that are sent across the connection. A node receiving data must acknowledge the arrival of each incoming packet so that the sending node is notified that the transmission was successful; lost packets must be retransmitted. However, data packets do not have to be acknowledged individually. Instead, most TCP implementations employ some delay mechanism that allows the TCP to acknowledge multiple incoming packets with a single acknowledgment and,

[*] Received by the editors March 4, 2004; accepted for publication (in revised form) June 3, 2005; published electronically December 30, 2005. A preliminary version of this paper appeared in *Proceedings of the* 14th *Annual ACM-SIAM Symposium on Discrete Algorithms*, Baltimore, MD, 2003, pp. 47–55.

http://www.siam.org/journals/sidma/19-4/44165.html

[†] Institut für Informatik, Albert-Ludwigs-Universität Freiburg, Georges-Köhler-Allee 79, 79110 Freiburg, Germany (salbers@informatik.uni-freiburg.de). This author's work was supported by the Deutsche Forschungsgemeinschaft, project AL 464/3-1,2, and by the EU, projects APPOL and APPOL II.

[‡] Department of Computer Science I, RWTH Aachen Ahornstr. 55, 52074 Aachen, Germany (bals@cs.rwth-aachen.de).

possibly, to piggyback the acknowledgment on an outgoing data segment. Reducing the number of acknowledgments has several advantages, e.g., the overhead incurred at the network nodes for sending and receiving acknowledgments is reduced and, more importantly, the network congestion is reduced. On the other hand, by reducing the number of acknowledgments, one adds latency to a TCP connection, which is not desirable. The goal is to balance the reduction in the number of acknowledgments with the increase in latency. The decision when to send acknowledgments must usually be made *online*, i.e., without knowledge of future packet arrival times.

Motivated by the fact that TCP supports dynamic acknowledgment mechanisms, Dooly, Goldman, and Scott [2, 3] formulated the following problem. A network node receives a sequence of n data packets. Let a_i denote the arrival time of packet i, $1 \leq i \leq n$. At time a_i, the arrival times a_j, $j > i$, are not known. We have to partition the sequence $\sigma = (a_1, \ldots, a_n)$ of packet arrival times into m subsequences $\sigma_1, \ldots, \sigma_m$, for some $m \geq 1$, such that each subsequence ends with an acknowledgment. We use σ_i to denote the set of arrivals in the partition. Let t_i be the time when the acknowledgment for σ_i is sent. We require $t_i \geq a_j$ for all $a_j \in \sigma_i$. If data packets are not acknowledged immediately, there are *acknowledgment delays*. Note that any reasonable objective function must take into account both the number of acknowledgments sent and the incurred acknowledgment delays. Ignoring the number of acknowledgments and considering only delays, it would be optimal to acknowledge each packet immediately, which leads to a large number of acknowledgments sent. On the other hand, ignoring delays and considering only acknowledgments, it would be best to send a single acknowledgment at the end of the packet sequence, which leads to unacceptable delays.

Previous results. Previous work on the dynamic TCP acknowledgment problem [2, 3, 4, 5, 6, 7] has focused mostly on the objective function that minimizes the number of acknowledgments and the sum of the delays incurred for all of the packets, i.e., we wish to minimize $h = m + \sum_{i=1}^{m} \sum_{a_j \in \sigma_i} (t_i - a_j)$. Given a solution generated by an acknowledgment algorithm A on input σ, the resulting objective function value is also referred to as the *cost* $C_A(\sigma)$ of A on σ. Following [8], an online algorithm A is called *c-competitive* if there exists a constant b such that $C_A(\sigma) \leq c \cdot C_{OPT}(\sigma) + b$ for all inputs σ. Here $C_{OPT}(\sigma)$ is the cost incurred by an optimal offline algorithm that knows the entire input σ in advance and can serve it with minimum cost.

Dooly, Goldman, and Scott [2, 3] presented a deterministic 2-competitive online algorithm and showed that no deterministic online strategy can achieve a smaller competitive ratio. This performance guarantee also holds if an online algorithm has some bounded lookahead. Most implementations of TCP have a *maximum delay constraint*, i.e., the acknowledgment of a packet may be delayed for at most δ time units, e.g., δ could be 500 ms. Dooly, Goldman, and Scott showed that their algorithm can be modified and remain 2-competitive in the presence of such a constraint. Karlin, Kenyon, and Randall [4] studied randomized online algorithms against oblivious adversaries. They developed a randomized online strategy that achieves a competitiveness of $e/(e-1) \approx 1.58$. Noga [5] and Seiden [7] independently showed that no randomized algorithm can do better.

Dooly, Goldman, and Scott also studied the minimization of a second objective function $h' = m + \sum_{i=1}^{m} \max_{a_j \in \sigma_i} (t_i - a_j)$, where one considers the sum of the maximum delays incurred in subsequences σ_i in addition to the number of acknowledgments sent. They showed that the best competitive ratio of a deterministic online algorithm without lookahead is equal to 2.

In general, Dooly, Goldman, and Scott and Karlin, Kenyon, and Randall pointed out that the TCP acknowledgment problem with objective functions h and h' are ski rental–type problems.

Our contribution. In this paper we investigate a new family of objective functions that penalize long acknowledgment delays of individual data packets more heavily. TCP is used for both interactive and bulk data transfer. In the first case, consider a TCP connection that is used for communication with a remote interactive program. Here, long delays are not acceptable as they are noticeable to the user. In the case of bulk data transfer, long delays also have a negative effect, and hence, as already mentioned before, most systems work with a maximum delay constraint. Short delays are of particular importance in time-critical applications. Therefore it is desirable to design algorithms that aim to keep the maximum delay short.

We study the objective function that minimizes the number of acknowledgments and the maximum delay incurred for any of the data packets. Given an input σ, consider a partitioning $\sigma_1, \ldots, \sigma_m$. Let $d_i = \max_{a_j \in \sigma_i}(t_i - a_j)$ be the maximum delay of any packet in σ_i, $1 \leq i \leq m$. We wish to minimize the function

$$f = m + \max_{1 \leq i \leq m} d_i. \tag{1.1}$$

It turns out that the dynamic TCP acknowledgment problem with objective function f is different in structure than the problem with functions h or h'. In particular our problem is not a ski rental problem. In section 2 we present a family of deterministic online algorithms and prove that the best strategy in that family achieves a competitive ratio of $\pi^2/6 \approx 1.644$. Note that $\pi^2/6 = \sum_{i=1}^{\infty} 1/i^2$. We also show that this is the best possible competitive ratio. No deterministic online algorithm can achieve a competitiveness smaller than $\pi^2/6$. Additionally, we investigate a generalization of the objective function f where delays are taken to the pth power and hence are penalized even more heavily. For any integer $p \geq 1$, we wish to minimize

$$f_p = m + \max_{1 \leq i \leq m} d_i^p. \tag{1.2}$$

For the formulation of the competitive ratio, let $\zeta(p) = \sum_{i=1}^{\infty} \frac{1}{i^p}$ for any $p \geq 2$. The function $\zeta(p)$ is known as the Riemann zeta function. We define $\zeta(1) := 1$. Let

$$c_p = 1 + \sum_{q=1}^{p+1} (-1)^{p+1-q} \zeta(q).$$

In section 3 we give a deterministic online algorithm that is c_p-competitive and prove that no deterministic strategy can achieve a competitiveness smaller than c_p. For $p = 1$, this expression is equal to $\pi^2/6$. In general, c_p is decreasing in p and tends to 1.5 as $p \to \infty$.

In section 4 we consider randomized online algorithms against oblivious adversaries and present lower bounds. We first prove that, given function f, no randomized online algorithm achieves a competitive ratio smaller than $3/(3 - 2/e) \approx 1.324$. We then show a lower bound of $2/(2 - 1/e) \approx 1.225$ for function f_p, $p \geq 2$.

We remark that, similar to [2, 3], we could consider in f and f_p a linear combination of the number of acknowledgments sent and the maximum delay (taken to the pth power). This does not change the competitive ratios, and the upper and lower bound proofs can be modified in a straightforward way. For simplicity, we study the functions as defined in (1.1) and (1.2).

2. Minimizing the maximum delay. In this section we study the objective function $f = m + \max_{1 \leq i \leq m} d_i$. We first present an online algorithm that achieves a competitiveness of $\pi^2/6$ and then develop a matching lower bound.

2.1. An optimal deterministic online algorithm. We define a general class of algorithms. Let z be a positive real number.

ALGORITHM LINEAR-DELAY(z). Initially, set $d = z$, and send the first acknowledgment at time $a_1 + d$. In general, suppose that the ith acknowledgment has just been sent and that j packets have been processed so far. Set $d = (i + 1)z$, and send the $(i + 1)$st acknowledgment at time $a_{j+1} + d$.

We analyze the algorithm for values z with $z \geq 1/2$ which give the best performance.

THEOREM 2.1. *For any z with $z \geq 1/2$, Linear-Delay(z) is c-competitive, where* $c = \max\{1 + z, (1 + z)/(2 + z - \pi^2/6)\}$.

COROLLARY 2.2. *Setting $z = \pi^2/6 - 1$, Linear-Delay(z) achieves a competitive ratio of $\pi^2/6$.*

We now prove Theorem 2.1.

Proof. In the following we call the online algorithm LD(z) for short. Suppose that LD(z) serves the input sequence using m acknowledgments. The longest acknowledgment delay is mz, and hence the online cost is $C_{LD(z)}(\sigma) = m(1 + z)$.

We have to lower bound the cost incurred by an optimum offline algorithm, OPT. In the sequence of n packets we identify a subsequence of m *main packets*, numbered from 0 to $m - 1$. Main packet 0 is the first packet in the input sequence. Main packet i, $1 \leq i \leq m - 1$, is the first packet that arrives after the ith acknowledgment sent by LD(z), i.e., it is the first packet that arrives after time t_i. The definition of LD(z) implies that the time difference between the $(i - 1)$st and the ith main packets is larger than iz for $i = 1, \ldots, m - 1$.

Suppose that the optimum offline algorithm serves the request sequence using l acknowledgments and that the maximum acknowledgment delay is C, $C \geq 0$. Then $C_{OPT}(\sigma) = l + C$. Associated with each acknowledgment α sent by OPT is an *acknowledgment interval* that starts when the first packet acknowledged by α arrives and ends when α is sent. The length of each interval is bounded by C. In the following i always denotes a positive integer.

LEMMA 2.3. *Any acknowledgment interval starting at or after the arrival of main packet $\lfloor \frac{C}{iz} \rfloor$ can contain at most i main packets.*

Proof. Main packet k with $k \geq \lfloor \frac{C}{iz} \rfloor + 1$ has a distance of more than $z(\lfloor \frac{C}{iz} \rfloor + 1)$ to the previous main packet. If the acknowledgment interval contained at least $i+1$ main packets, then the length of the interval would be at least $iz(\lfloor \frac{C}{iz} \rfloor + 1) > iz(\frac{C}{iz}) = C$, which is impossible. □

Define $i_0 = \lfloor \sqrt[3]{C/z} \rfloor - 1$. In the rest of this proof we assume $i_0 \geq 2$. If $i_0 \leq 1$, then $C \leq 27z$ and OPT must acknowledge each of the last $m - 27$ main packets with separate acknowledgments. In this case LD(z) is clearly $(1 + z)$-competitive.

LEMMA 2.4. *Let $1 \leq i \leq i_0$. The acknowledgment interval containing main packet k, for $k \geq \lfloor \frac{C}{iz} \rfloor$, must have started after the arrival of main packet $\lfloor \frac{C}{(i+1)z} \rfloor$.*

Proof. We show that the time window starting at main packet $\lfloor \frac{C}{(i+1)z} \rfloor$ and ending with main packet $\lfloor \frac{C}{iz} \rfloor$ is larger than C, which proves the lemma. The number of main packets in this time window is

$$\lfloor \tfrac{C}{iz} \rfloor - \lfloor \tfrac{C}{(i+1)z} \rfloor + 1 > \tfrac{C}{i(i+1)z} \geq i + 2.$$

The last inequality holds because it is equivalent to $C/z \geq i(i + 1)(i + 2)$, and this is satisfied for all $i \leq i_0$. Thus there are at least $i + 2$ main packets in this time window. Each of the last $i + 1$ of these is more than $z(\lfloor \frac{C}{(i+1)z} \rfloor + 1)$ time units away from the previous main packet, and thus the length of the window is greater than $(i+1)z(\lfloor \frac{C}{(i+1)z} \rfloor + 1) > (i+1)z(\frac{C}{(i+1)z}) = C$. □

We now lower bound the number of acknowledgments sent by OPT and use the following charging scheme. An acknowledgment costs 1. We charge this cost to the main packets contained in the associated acknowledgment interval and split the cost evenly among these main packets. More specifically, if an acknowledgment interval contains $i \geq 1$ main packets, then each of these packets is assigned a cost of $1/i$. If an acknowledgment interval does not contain a main packet, then we ignore it in the analysis of OPT's cost. We develop a lower bound on the cost charged to each main packet. Summing over all main packets, we derive a lower bound on the optimum cost incurred for sending acknowledgments.

We assume that $C < m$. If $C \geq m$, then LD(z) is clearly $(1+z)$-competitive because LD(z)'s cost is $(1+z)m$ and the optimum offline cost is at least m. In the following we will first analyze the case $C \leq zm$ and then $C > zm$.

Suppose that $C \leq zm$. Each main packet is contained in some acknowledgment interval. Let i be an integer with $1 \leq i \leq i_0$. We analyze the cost charged to main packet k with $k \geq \lfloor \frac{C}{iz} \rfloor$ and $k < \lfloor \frac{C}{(i-1)z} \rfloor$. If $i = 1$, then the expression $\lfloor \frac{C}{(i-1)z} \rfloor$ is undefined, and we have the trivial upper bound $k < m$. Consider an arbitrary i with $1 \leq i \leq i_0$. If the acknowledgment interval containing main packet k started at or after the arrival of main packet $\lfloor \frac{C}{iz} \rfloor$, then by Lemma 2.3 at most i main packets are contained in the interval, and main packet k is assigned a cost of at least $1/i$. If the acknowledgment interval started earlier, then by Lemma 2.4 it must have started after the arrival of main packet $\lfloor \frac{C}{(i+1)z} \rfloor$. Applying Lemma 2.3 for $i+1$, we find that the interval contains at most $i+1$ main packets and the packet is assigned a cost of at least $1/(i+1)$. There is only one acknowledgment interval that starts before and ends after the arrival of main packet $\lfloor \frac{C}{iz} \rfloor$. Thus for at most $i+1$ main packets considered above, the cost is lower bounded by $1/(i+1)$ instead of $1/i$. We find that, for $2 \leq i \leq i_0$, the total cost assigned to main packets k with $\lfloor \frac{C}{iz} \rfloor \leq k < \lfloor \frac{C}{(i-1)z} \rfloor$ is at least

$$(\lfloor \tfrac{C}{(i-1)z} \rfloor - \lfloor \tfrac{C}{iz} \rfloor)\tfrac{1}{i} - (i+1)(\tfrac{1}{i} - \tfrac{1}{i+1}) = (\lfloor \tfrac{C}{(i-1)z} \rfloor - \lfloor \tfrac{C}{iz} \rfloor)\tfrac{1}{i} - \tfrac{1}{i}.$$

For $i = 1$, the total cost assigned to all the main packets k with $k \geq \lfloor \frac{C}{z} \rfloor$ is

$$(m - \lfloor \tfrac{C}{z} \rfloor) - 2(1 - \tfrac{1}{2}) = (m - \lfloor \tfrac{C}{z} \rfloor) - 1.$$

Summing over all i, we find that the number of acknowledgments sent by OPT is at least

$$\begin{aligned} l &\geq m - \lfloor \tfrac{C}{z} \rfloor - 1 + \sum_{i=2}^{i_0} \left(\left(\left\lfloor \tfrac{C}{(i-1)z} \right\rfloor - \lfloor \tfrac{C}{iz} \rfloor \right) \tfrac{1}{i} - \tfrac{1}{i} \right) \\ &= m - \sum_{i=1}^{i_0 - 1} \lfloor \tfrac{C}{iz} \rfloor \left(\tfrac{1}{i} - \tfrac{1}{i+1} \right) - \left\lfloor \tfrac{C}{i_0 z} \right\rfloor \tfrac{1}{i_0} - H_{i_0}. \end{aligned}$$

Here H_{i_0} denotes the i_0th Harmonic number. Thus

$$\begin{aligned} l &\geq m - \tfrac{C}{z} \sum_{i=1}^{\infty} \left(\tfrac{1}{i^2} - \tfrac{1}{i(i+1)} \right) - \tfrac{C}{i_0^2 z} - i_0 \\ &\geq m - \tfrac{C}{z} \left(\tfrac{\pi^2}{6} - 1 \right) - \tfrac{C}{i_0^2 z} - i_0 \end{aligned}$$

$$\begin{aligned}
&\geq m - \tfrac{C}{z}\left(\tfrac{\pi^2}{6} - 1\right) - 10\sqrt[3]{\tfrac{C}{z}} \\
&> m - \tfrac{C}{z}\left(\tfrac{\pi^2}{6} - 1\right) - 10\sqrt[3]{\tfrac{m}{z}}.
\end{aligned}$$

The second-to-last inequality follows because $i_0 \geq 2$, and hence $i_0^2 \geq \frac{1}{9}(\frac{C}{z})^{2/3}$. The last inequality follows because $C < m$, and hence $\sqrt[3]{C/z} < \sqrt[3]{m/z}$. The total cost incurred by OPT is at least

$$C_{OPT}(\sigma) = l + C \geq m + C - \tfrac{C}{z}\left(\tfrac{\pi^2}{6} - 1\right) - O\left(\sqrt[3]{m}\right). \tag{2.1}$$

We now distinguish two cases. If $z > \frac{\pi^2}{6} - 1$, then the right-hand side of (2.1) is increasing in C. As we have to lower bound $C_{OPT}(\sigma)$, choosing $C = 0$ we obtain $C_{OPT}(\sigma) \geq m - O(\sqrt[3]{m})$. This implies that LD($z$) is $(1 + z)$-competitive because the online cost is $(1 + z)m$. If $z \leq \frac{\pi^2}{6} - 1$, then the right-hand side of (2.1) is decreasing in C. Choosing the largest possible value $C = zm$, we obtain $C_{OPT}(\sigma) \geq (2+z-\frac{\pi^2}{6})m - O(\sqrt[3]{m})$, and LD($z$) achieves a competitive ratio of $(1+z)/(2+z-\frac{\pi^2}{6})$.

We next analyze the case $C > zm$. The only difference in analyzing this case is that there are no main packets k with $k \geq \lfloor \frac{C}{z} \rfloor$ and $k < m$ because C is large. However, there are main packets k with $k \geq \lfloor \frac{C}{2z} \rfloor$ and $k < m$ because $C < m \leq 2zm$ since $z \geq 1/2$. Thus the number of acknowledgments sent by OPT is

$$\begin{aligned}
l &\geq \left(m - \lfloor \tfrac{C}{2z} \rfloor\right) \tfrac{1}{2} - \tfrac{1}{2} + \sum_{i=3}^{i_0} \left(\left(\left\lfloor \tfrac{C}{(i-1)z} \right\rfloor - \lfloor \tfrac{C}{iz} \rfloor \right) \tfrac{1}{i} - \tfrac{1}{i} \right) \\
&\geq \tfrac{1}{2}m - \sum_{i=2}^{i_0-1} \tfrac{C}{iz}\left(\tfrac{1}{i} - \tfrac{1}{i+1}\right) - \tfrac{C}{i_0^2 z} - (H_{i_0} - 1) \\
&\geq \tfrac{1}{2}m - \tfrac{C}{z}\left(\tfrac{\pi^2}{6} - 1.5\right) - 10\sqrt[3]{\tfrac{m}{z}}.
\end{aligned}$$

Thus the optimum cost is at least $C_{OPT}(\sigma) \geq \frac{1}{2}m + C - \frac{C}{z}(\frac{\pi^2}{6} - 1.5) - O(\sqrt[3]{m})$. The right-hand side of the last inequality is increasing in C because $z \geq 1/2 > (\frac{\pi^2}{6} - 1.5)$. Since $C > zm$, we obtain $C_{OPT}(\sigma) \geq (2 + z - \frac{\pi^2}{6})m - O(\sqrt[3]{m})$, and LD($z$) achieves a competitive ratio of $(1 + z)/(2 + z - \frac{\pi^2}{6})$. □

2.2. Lower bound.

THEOREM 2.5. *Let A be a deterministic online algorithm. If A is c-competitive, then $c \geq \frac{\pi^2}{6}$.*

Proof. We construct a family of request sequences σ_l for any $l \geq 8$. For a fixed l in this range, let $i_0 = \lfloor \sqrt[3]{l} \rfloor - 2$ and $l' = \lfloor \frac{l}{i_0+1} \rfloor$. For convenience we number the packets in σ_l starting with l'. Packet l' is sent at time 0. For any i with $l' < i \leq l$, packet i is sent exactly $(\pi^2/6 - 1)i$ time units after packet $i - 1$. For any i with $i > l$, packet i is sent exactly $(\pi^2/6 - 1)l$ time units after packet $i - 1$. The adversary stops sending packets as soon as the online algorithm decides to acknowledge an incoming packet together with the preceding packet. If the online algorithm never acknowledges a packet together with a preceding packet, the adversary can force a competitive ratio arbitrarily close to 2 by always acknowledging two packets together. Thus, let m be the number of the last packet sent by the adversary. Note that m is a function of l but, for simplicity, this dependency will not be shown in the notation.

In the following we will first analyze the competitive ratio of the online algorithm if $m \le l$. Then we will consider the case $m > l$. If $m \le l$, then the adversary can acknowledge each incoming packet immediately, and its cost is $C_{ADV}(\sigma_l) = m - l' + 1$ because the packet numbering in the sequence starts with l'. The online algorithm A serves the first $m - l' - 1$ packets with separate acknowledgments and the last two packets with a joint acknowledgment. The acknowledgment of packet $m-1$ is delayed by $(\frac{\pi^2}{6} - 1)m$ time units. Thus the total online cost is at least

$$\begin{aligned} C_A(\sigma_l) &= m - l' + (\tfrac{\pi^2}{6} - 1)m \\ &= \tfrac{\pi^2}{6} m - l' \\ &= \tfrac{\pi^2}{6}(m - \tfrac{6}{\pi^2} l') \\ &\ge \tfrac{\pi^2}{6}(m - l' + 1) \\ &= \tfrac{\pi^2}{6} C_{ADV}(\sigma_l). \end{aligned}$$

The last inequality holds because $l' \ge \frac{\pi^2}{6}/(\frac{\pi^2}{6} - 1)$. To verify this relation we observe that, for $l = 8$, $l' = 8 \ge \frac{\pi^2}{6}/(\frac{\pi^2}{6} - 1)$ and l' is increasing in l.

It remains to analyze the case $m > l$. The adversary chooses acknowledgment intervals of length $(\frac{\pi^2}{6} - 1)l$; i.e., it sends out an acknowledgment whenever there is an unacknowledged packet waiting for exactly $(\frac{\pi^2}{6} - 1)l$ time units. To analyze the number of acknowledgments incurred by the adversary, we need the following lemma.

LEMMA 2.6. *Let $1 \le i \le i_0$. An acknowledgment interval that ends after the arrival of packet $\lfloor \frac{l}{i} \rfloor$ must have started after the arrival of packet $\lfloor \frac{l}{i+1} \rfloor$.*

Proof. Suppose that an acknowledgment interval ending after the arrival of packet $\lfloor \frac{l}{i} \rfloor$ started at or before the arrival of packet $\lfloor \frac{l}{i+1} \rfloor$. This time interval contains $\lfloor \frac{l}{i} \rfloor - \lfloor \frac{l}{i+1} \rfloor$ packets that are at least $(\frac{\pi^2}{6} - 1)(\lfloor \frac{l}{i+1} \rfloor + 1) \ge (\frac{\pi^2}{6} - 1)\frac{l}{i+1}$ time units away from the preceding packet. Thus the time interval has a total length of

$$(\tfrac{\pi^2}{6} - 1)\tfrac{l}{i+1}(\lfloor \tfrac{l}{i} \rfloor - \lfloor \tfrac{l}{i+1} \rfloor) \ge (\tfrac{\pi^2}{6} - 1)\tfrac{l}{i+1}(\tfrac{l}{i} - 1 - \tfrac{l}{i+1}) = (\tfrac{\pi^2}{6} - 1)\tfrac{l}{i+1}(\tfrac{l}{i(i+1)} - 1).$$

We have $\frac{l}{i(i+1)} - 1 > i+1$ because the this inequality is equivalent to $l > i(i+1)(i+2)$, which holds for all $1 \le i \le i_0$. Thus the time interval has a total length of greater than $(\frac{\pi^2}{6} - 1)l$, contradicting the fact that the adversary chooses acknowledgment intervals of length $(\frac{\pi^2}{6} - 1)l$. □

To upper bound the total number of acknowledgments incurred by the adversary, we use a charging scheme similar to that employed in the upper bound. If an acknowledgment interval contains i packets, then the cost of 1 is distributed evenly among the packets, i.e., each packet is assigned a cost of $\frac{1}{i}$. An acknowledgment interval that ends no later than the arrival of packet $\lfloor \frac{l}{i} \rfloor, 1 \le i \le i_0$, contains at least $i + 1$ packets because each of the packets is a distance of at most $\lfloor \frac{l}{i} \rfloor (\frac{\pi^6}{6} - 1)$ away from the preceding packet. Hence packets k with $\lfloor \frac{l}{i+1} \rfloor < k \le \lfloor \frac{l}{i} \rfloor$ are charged a cost of at most $\frac{1}{i+1}$. However, this is not completely correct because a packet k in the latter range may be contained in an acknowledgment interval that ends after the arrival of packet $\lfloor \frac{l}{i} \rfloor$. By the above lemma, such an acknowledgment interval cannot end after the arrival of packet $\lfloor \frac{l}{i-1} \rfloor$ if $i \ge 2$. Thus the packet k is assigned a cost of $\frac{1}{i}$ instead of $\frac{1}{i+1}$. At most $i + 1$ packets can have this slightly higher cost because each packet k with $\lfloor \frac{l}{i+1} \rfloor < k \le \lfloor \frac{l}{i} \rfloor$ has a distance of at least $(\frac{\pi^2}{6} - 1)(\lfloor \frac{l}{i+1} \rfloor + 1) > (\frac{\pi^2}{6} - 1)\frac{l}{i+1}$

to its preceding packet. For any $1 \le i \le i_0$, the total cost charged to all the packets k with $\lfloor \frac{l}{i+1} \rfloor < k \le \lfloor \frac{l}{i} \rfloor$ is

$$(\lfloor \tfrac{l}{i} \rfloor - \lfloor \tfrac{l}{i+1} \rfloor)\tfrac{1}{i+1} + (i+1)(\tfrac{1}{i} - \tfrac{1}{i+1}) = (\lfloor \tfrac{l}{i} \rfloor - \lfloor \tfrac{l}{i+1} \rfloor)\tfrac{1}{i+1} + \tfrac{1}{i}.$$

Any packet k with $l < k \le m$ is charged a cost of $1/2$ because these packets are a distance of exactly $(\frac{\pi^2}{6} - 1)l$ apart. In the worst case, the last packet m is charged a cost of 1. Moreover packet l' is assigned a cost of $\frac{1}{i_0+1}$. In summary, the total cost charged to all of the packets, which is equal to the total number of acknowledgments sent by the adversary, is upper bounded by

$$\begin{aligned}
&(m-l-1)\tfrac{1}{2} + 1 + \sum_{i=1}^{i_0} \left(\left(\left\lfloor \tfrac{l}{i} \right\rfloor - \left\lfloor \tfrac{l}{i+1} \right\rfloor \right) \tfrac{1}{i+1} + \tfrac{1}{i} \right) + \tfrac{1}{i_0+1} \\
&\le \tfrac{m}{2} - \tfrac{l}{2} + \tfrac{1}{2} + \sum_{i=1}^{i_0} \left(\tfrac{l}{i} - \tfrac{l}{i+1} + 1 \right) \tfrac{1}{i+1} + H_{i_0+1} \\
&\le \tfrac{m}{2} - \tfrac{l}{2} + \sum_{i=1}^{\infty} \tfrac{l}{i(i+1)} - \sum_{i=1}^{\infty} \tfrac{l}{(i+1)^2} + 2H_{i_0+1} \\
&= \tfrac{m}{2} + \tfrac{l}{2} - l \left(\tfrac{\pi^2}{6} - 1 \right) + O(\log l),
\end{aligned}$$

where H_k is the kth Harmonic number.

Since the maximum acknowledgment delay incurred by the adversary is $(\frac{\pi^2}{6} - 1)l$, its total cost is $\frac{1}{2}(m+l) + O(\log l)$. On the other hand, the total cost incurred by the online algorithm A is $m - l' + (\frac{\pi^2}{6} - 1)l$ because the input consists of $m - l' + 1$ data packets, the last two of which are acknowledged together. We conclude that the ratio of the online cost to the adversary's cost is

$$\frac{\frac{\pi^2}{6}l + m - l - l'}{l + \frac{1}{2}(m-l) + O(\log l)}.$$

Since $l' = o(l)$ and $O(\log l) = o(l)$, this ratio approaches a value of at least $\frac{\pi^2}{6}$ as $l \to \infty$, no matter how the online algorithm chooses $m, m > l$. □

3. Minimizing the maximum delay taken to the *p*th power. In this section we study the general objective function $f_p = m + \max_{1 \le i \le m} d_i^p$ and show that $c_p = 1 + \sum_{q=1}^{p+1} (-1)^{p+1-q} \zeta(q)$ is the best competitiveness of deterministic online algorithms. Before we give the upper and lower bound analyses, we briefly analyze c_p. We show that it is decreasing in p and tends to 1.5 as $p \to \infty$. For $p \ge 1$, let $g(p) = \sum_{i=1}^{\infty} \frac{1}{i^p(i+1)}$. Then, for $p \ge 2$,

$$\begin{aligned}
g(p) &= \sum_{i=1}^{\infty} \frac{1}{i^p(i+1)} = \sum_{i=1}^{\infty} \frac{1}{i^p} - \sum_{i=1}^{\infty} \frac{1}{i^{p-1}(i+1)} \\
&= \zeta(p) - g(p-1).
\end{aligned}$$

Applying this recurrence repeatedly we obtain $g(p) = \sum_{q=1}^{p} (-1)^{p-q} \zeta(q)$. Note that $g(1) = 1 = \zeta(1)$. Thus $c_p = 1 + g(p+1)$. We have $g(p+1) = \frac{1}{2} + \sum_{i=2}^{\infty} \frac{1}{i^{p+1}(i+1)}$. The last sum is always positive and tends to 0 as $p \to \infty$. Table 3.1 shows the value of c_p for small p.

TABLE 3.1
Some values of c_p.

p	c_p
1	1.6449
2	1.5571
3	1.5252
4	1.5117
5	1.5056
6	1.5027
7	1.5013
8	1.5007
9	1.5003
10	1.5002

3.1. An optimal deterministic online algorithm. We generalize the algorithm given in section 2. Let z be a positive real number.

ALGORITHM DELAY(z,p). Set the initial delay to $d = \sqrt[p]{z}$, and send out the first acknowledgment at time $a_1 + d$. In general, assume that i acknowledgments have been sent and that j packets have been processed so far. Set $d = \sqrt[p]{(i+1)z}$, and send the $(i+1)$st acknowledgment at time $a_{j+1} + d$.

THEOREM 3.1. *Setting $z_p = c_p - 1$, the algorithm Delay(z_p, p) is c_p-competitive.*

Proof. We denote the algorithm by $\mathrm{D}(z_p, p)$ for short. Suppose that the online algorithm serves the input sequence using m acknowledgments. Then its total cost is $C_{D(z_p,p)}(\sigma) = m + (\sqrt[p]{m\, z_p})^p = (1+z_p)m = c_p m$. Let C be the maximum acknowledgment delay incurred by the optimum offline algorithm OPT. If $C > \sqrt[p]{m}$, then the optimum offline cost is at least m, and $\mathrm{D}(z_p, p)$ is clearly c_p-competitive. Therefore we assume $C \le \sqrt[p]{m}$.

In analyzing the optimum offline cost, we use the terms *main packet* and *acknowledgment interval* as introduced in the proof of Theorem 2.1. Again we number the m main packets in the input from 0 to $m-1$. Let $i_0 = \lfloor \sqrt[2p+1]{C^p/z_p} \rfloor - 1$. In the following we assume $i_0 \ge 4$. If $i_0 \le 3$, then $\sqrt[2p+1]{C^p/z_p} \le 5$, which is equivalent to $C \le \sqrt[p]{5^{2p+1} z_p}$. Thus C is upper bounded by a constant, and all but a constant number of the m main packets require a separate acknowledgment by OPT. Thus $\mathrm{D}(z_p, p)$ is c_p-competitive.

In the following we first concentrate on the case $C < \sqrt[p]{z_p m}$, then we consider $C \ge \sqrt[p]{z_p m}$. To lower bound the number of acknowledgments sent by OPT, we apply the usual charging scheme. If an acknowledgment interval contains i main packets, we charge a cost of $\frac{1}{i}$ to each of these. Using ideas similar to that in the proof of Theorem 2.1, we can show that an acknowledgment interval starting at or after the arrival of main packet $\lfloor \frac{C^p}{i^p z_p} \rfloor$ can contain at most i main packets. Second, an acknowledgment interval containing main packet k with $k \ge \lfloor \frac{C^p}{i^p z_p} \rfloor$ must have started after packet $\lfloor \frac{C^p}{(i+1)^p z_p} \rfloor$. These two statements imply that the total cost assigned to main packets k with $k \ge \lfloor \frac{C^p}{z_p} \rfloor$ is at least $m - \lfloor \frac{C^p}{z_p} \rfloor - 1$ and that the cost assigned to main packets k with $\lfloor \frac{C^p}{i^p z_p} \rfloor \le k < \lfloor \frac{C^p}{(i-1)^p z_p} \rfloor$ and $2 \le i \le i_0$ is at least

$$\left(\left\lfloor \frac{C^p}{(i-1)^p z_p} \right\rfloor - \left\lfloor \frac{C^p}{i^p z_p} \right\rfloor \right) \frac{1}{i} - \frac{1}{i}.$$

Hence the number l of acknowledgments sent by OPT is at least

$$\begin{aligned} l &\geq m - \left\lfloor \tfrac{C^p}{z_p} \right\rfloor - 1 + \sum_{i=2}^{i_0} \left(\left(\left\lfloor \tfrac{C^p}{(i-1)^p z_p} \right\rfloor - \left\lfloor \tfrac{C^p}{i^p z_p} \right\rfloor \right) \tfrac{1}{i} - \tfrac{1}{i} \right) \\ &= m - \sum_{i=1}^{i_0-1} \left\lfloor \tfrac{C^p}{i^p z_p} \right\rfloor \left(\tfrac{1}{i} - \tfrac{1}{i+1} \right) - \left\lfloor \tfrac{C^p}{i_0^p z_p} \right\rfloor \tfrac{1}{i} - H_{i_0} \\ &\geq m - \tfrac{C^p}{z_p} \sum_{i=1}^{\infty} \left(\tfrac{1}{i^{p+1}} - \tfrac{1}{i^p(i+1)} \right) - \tfrac{C^p}{i_0^{p+1} z_p} - H_{i_0} \\ &= m - \tfrac{C^p}{z_p} (\zeta(p+1) - g(p)) - \tfrac{C^p}{i_0^{p+1} z_p} - H_{i_0} \\ &= m - \tfrac{C^p}{z_p} z_p - \tfrac{C^p}{i_0^{p+1} z_p} - H_{i_0}. \end{aligned}$$

The last equation holds because $\zeta(p+1) - g(p) = g(p+1) = c_p - 1 = z_p$.

LEMMA 3.2. *The term* $\frac{C^p}{i_0^{p+1} z_p}$ *is* $o(m)$.

Proof. By definition the term equals

$$\frac{C^p}{\left(\left\lfloor \sqrt[2p+1]{\frac{C^p}{z_p}} \right\rfloor - 1 \right)^{p+1} z_p} \leq \frac{C^p}{\left(\sqrt[2p+1]{\frac{C^p}{z_p}} - 2 \right)^{p+1} z_p}, \tag{3.1}$$

where the inequality holds because the denominator is positive by the choice of $i_0 \geq 4$. Moreover, the assumption $i_0 \geq 4$ implies $2 \leq i_0/2 \leq \sqrt[2p+1]{C^p/z_p}/2$, and hence the last expression in (3.1) can be upper bounded by

$$\frac{C^p}{\left(\sqrt[2p+1]{\frac{C^p}{z_p}} - \sqrt[2p+1]{\frac{C^p}{z_p}}/2 \right)^{p+1} z_p} = C^{p - \frac{p(p+1)}{2p+1}} z_p^{\frac{p+1}{2p+1} - 1} 2^{p+1} \ = \ dC^{p - \frac{p(p+1)}{2p+1}}$$

for some constant d. As $C \leq \sqrt[p]{m}$ and the exponent $p - \frac{p(p+1)}{2p+1}$ is strictly smaller than p, the term under consideration is $o(m)$. □

Using the above lemma we find that the number of acknowledgments sent by OPT is at least $m - C^p - o(m)$ and that the total cost is $C_{OPT}(\sigma) \geq m - o(m)$. This implies that $\mathrm{D}(z_p, p)$ is c_p-competitive.

We finally analyze the case $C \geq \sqrt[p]{z_p m}$. Since C is large, there are not necessarily main packets k with $k \geq \lfloor \frac{C^p}{z_p} \rfloor$ and $C < m$. However, there are packets $k \geq \lfloor \frac{C^p}{2^p z_p} \rfloor$ and $k < m$ because $C^p/(2^p z_p) < m$ is equivalent to $C < \sqrt[p]{m 2^p z_p}$, and this holds because $C \leq \sqrt[p]{m}$ and $z_p = g(p+1) = \frac{1}{2} + \sum_{i=2}^{\infty} \frac{1}{i^{p+1}(i+1)} > \frac{1}{2}$. Thus the number of acknowledgments l sent by OPT is at least

$$\begin{aligned} l &\geq \left(m - \left\lfloor \tfrac{C^p}{2^p z_p} \right\rfloor \right) \tfrac{1}{2} - \tfrac{1}{2} + \sum_{i=3}^{i_0} \left(\left(\left\lfloor \tfrac{C^p}{(i-1)^p z_p} \right\rfloor - \left\lfloor \tfrac{C^p}{i^p z_p} \right\rfloor \right) \tfrac{1}{i} - \tfrac{1}{i} \right) \\ &\geq \tfrac{m}{2} - \tfrac{C^p}{z_p} \sum_{i=2}^{\infty} \left(\tfrac{1}{i^{p+1}} - \tfrac{1}{i^p(i+1)} \right) - \tfrac{C^p}{i_0^{p+1} z_p} - H_{i_0} \\ &= \tfrac{m}{2} - \tfrac{C^p}{z_p} (z_p - \tfrac{1}{2}) - o(m). \end{aligned}$$

We conclude that the optimum offline cost is at least $C_{OPT}(\sigma) \geq \frac{m}{2} + \frac{C^p}{2z_p} - o(m) \geq m - o(m)$ because $C \geq \sqrt[p]{z_p m}$, and $\mathrm{D}(z_p, p)$ is c_p-competitive. □

3.2. Lower bound.

THEOREM 3.3. *Let A be a deterministic online algorithm. If A is c-competitive, then $c \geq c_p$.*

Proof. We construct a family of input sequences σ_l for any integer $l \geq 1$. For a fixed l, let $i_0 = \lfloor \sqrt[2p+1]{l} \rfloor - 1$ and $l' = \lfloor \frac{l}{(i_0+1)^p} \rfloor$. Note that $l' = \Theta(l^{1-\frac{p}{2p+1}}) = o(l)$. We number the packets in σ_l starting with l'. Packet l' is sent at time 0. Packet k, for $l' < k \leq l$, is sent $\sqrt[p]{z_p k}$ time units after packet $k-1$. For $k > l$, packets k and $k-1$ are separated by exactly $\sqrt[p]{z_p l}$ time units. The adversary stops sending packets when the online algorithm decides to acknowledge two packets with the same acknowledgment. Let m be the number of the last packet sent.

If $m \leq l$, the cost incurred by the online algorithm A is at least $m - l' + (z_p m)^{\frac{p}{p}} = c_p m - l'$. The adversary can acknowledge each packet immediately, incurring no delays, so that its cost is at most $m - l' + 1$. The ratio of the cost incurred by A to the cost incurred by the adversary is at least

$$\frac{c_p m - l'}{m - l' + 1} = c_p + \frac{z_p l' - c_p}{m - l' + 1},$$

and this expression is at least c_p if $l \geq 2^{2p+1}$. In this case $l' \geq 4$ and $z_p l' - c_p \geq 0$ because $z_p > 1/2$ and $c_p < 2$.

We concentrate on the case $m > l$. The adversary chooses an acknowledgment interval of $\sqrt[p]{z_p l}$ time units. Using the familiar charging scheme we can show that, in order to upper bound the number of acknowledgments incurred by the adversary, the total cost charged to packets k with $\lfloor \frac{l}{i^p} \rfloor < k \leq \lfloor \frac{l}{(i+1)^p} \rfloor$ and $1 \leq i \leq i_0$ is at most $(\lfloor \frac{l}{i^p} \rfloor - \lfloor \frac{l}{(i+1)^p} \rfloor)\frac{1}{i+1} + \frac{1}{i}$. The total cost charged to packets k with $k > l$ is at most $(m - l)\frac{1}{2} + \frac{1}{2}$. Hence the total number of acknowledgments sent by the adversary is at most

$$\begin{aligned}
&(m-l)\tfrac{1}{2} + \tfrac{1}{2} + \sum_{i=1}^{i_0}\left(\left(\lfloor \tfrac{l}{i^p} \rfloor - \left\lfloor \tfrac{l}{(i+1)^p} \right\rfloor\right) \tfrac{1}{i+1} + \tfrac{1}{i}\right) + \tfrac{1}{i_0+1} \\
&\leq (m-l)\tfrac{1}{2} + \tfrac{l}{2} - \sum_{i=2}^{\infty} \frac{l}{i^{p+1}(i+1)} + 2H_{i_0+1} \\
&= (m-l)\tfrac{1}{2} + l - z_p l + O(\log l).
\end{aligned}$$

The total cost paid by the adversary is at most $(m-l)\frac{1}{2} + l + O(\log l)$, and the ratio of the cost incurred by A to the cost incurred by the adversary is at least

$$\frac{c_p l + m - l - l'}{l + \frac{1}{2}(m-l) + O(\log l)}.$$

This ratio approaches a value of at least c_p as $l \to \infty$ because $l' = o(l)$. □

4. Randomization. In this section we develop lower bounds on the competitive ratio achieved by randomized online algorithms.

THEOREM 4.1. *For the dynamic TCP acknowledgment problem with objective function f, no randomized online algorithm can achieve a competitive ratio smaller than $c \geq 3/(3 - \frac{2}{e})$ against any oblivious adversary.*

Proof. We apply Yao's principle [1, 9] and construct a probability distribution on input sequences σ_l, for any integer $l \geq 1$, such that, for any deterministic online algorithm D,

$$\lim_{l \to \infty} \frac{E[C_D(\sigma_l)]}{E[C_{ADV}(\sigma_l)]} \geq \frac{3}{3 - 2/e}$$

and

$$\lim_{l\to\infty} E[C_{ADV}(\sigma_l)] = \infty.$$

Here $E[C_{ADV}(\sigma_l)]$ and $E[C_D(\sigma_l)]$ denote the expected costs incurred by the adversary and the deterministic online algorithm, respectively. An input σ_l consists of *triples*. A triple is a set of three data packets that are separated by l time units each. More precisely, the second packet is sent exactly l time units after the first packet of the triple; the third packet is sent l time units after the second packet. Thus a triple has a total length of $2l$. The adversary sends triples, where the distance between triples is chosen to be so large that it does not make sense to acknowledge packets in two different triples with one acknowledgment. With probability $p_i = q(1-q)^{i-1}$, where $q = 1/l$, the adversary sends exactly i triples for any $i \geq 1$. Note that $\sum_{i=1}^{\infty} q(1-q)^{i-1} = 1$. Triple i and $i+1$ are separated by $3l/p_{i+1}$ time units.

If a deterministic online algorithm on this input acknowledges packets of different triples together and if this happens for the first time for packets from triples i and $i+1$, then the expected cost of the algorithm is at least $p_{i+1}(3l/p_{i+1}) = 3l$. In the following we concentrate on the case that a deterministic online algorithm on this input never acknowledges packets from different triples together. We characterize an algorithm by two nonnegative integers l_1, l_2, with $l_1 < l_2$ such that l_1+1 is the first triple where the algorithm acknowledges at least two packets together and l_2+1 is the index of the first triple where all three packets are acknowledged together. We refer to this strategy as $D(l_1, l_2)$, $l_1 \leq l_2$. Algorithm $D(l_1, \infty)$, $l_1 \geq 0$, never acknowledges all the packets of one triple together, and $D(\infty, \infty)$ never acknowledges any packets together. To analyze the expected cost, we need the following lemma.

LEMMA 4.2.

(a) *If* $l_1 < l_2$, *then* $E[C_{D(l_1,l_2)}(\sigma_l)] = E[C_{D(l_1+1,l_2)}(\sigma_l)]$.

(b) *If* $l_1 \leq l_2$, *then* $E[C_{D(l_1,l_2)}(\sigma_l)] = E[C_{D(l_1,l_2+1)}(\sigma_l)]$.

(c) *For any* $l_1 \geq 0$, $E[C_{D(l_1,\infty)}(\sigma_l)] = E[C_{D(l_1+1,\infty)}(\sigma_l)]$.

Proof. We prove part (a). The other parts can be proved in a similar manner. We have

$$\begin{aligned}
E[C_{D(l_1,l_2)}(\sigma_l)] &\geq \sum_{i=1}^{l_1} 3ip_i + \sum_{i=l_1+1}^{l_2} (l+l_1+2i)p_i + \sum_{i=l_2+1}^{\infty} (2l+l_1+l_2+i)p_i \\
&= \sum_{i=1}^{l_1+1} 3ip_i - 3(l_1+1)p_{l_1+1} + \sum_{l_1+2}^{l_2} (l+l_1+1+2i)p_i \\
&\quad +(l+l_1+2(l_1+1))p_{l_1+1} - \sum_{i=l_1+2}^{l_2} p_i \\
&\quad + \sum_{i=l_2+1}^{\infty} (2l+l_1+1+l_2+i)p_i - \sum_{i=l_2+1}^{\infty} p_i \\
&= E[C_{D(l_1+1,l_2)}(\sigma_l)] + (l-1)p_{l_1+1} - \sum_{i=l_1+2}^{\infty} p_i \\
&= E[C_{D(l_1+1,l_2)}(\sigma_l)]. \qquad \square
\end{aligned}$$

Parts (a) and (b) of Lemma 4.2 imply that $E[C_{D(0,0)}(\sigma_l)] = E[C_{D(l_1,l_2)}(\sigma_l)]$ for any $0 \leq l_1 \leq l_2$. Hence it suffices to compute $E[C_{D(0,0)}(\sigma_l)] = \sum_{i=1}^{\infty}(2l+i)p_i = 2l + 1/q = 3l$. Part (c) of Lemma 4.2 implies $E[C_{D(0,\infty)}(\sigma_l)] = E[C_{D(l_1,\infty)}(\sigma_l)]$ for any $l_1 \geq 0$. We have $E[C_{D(0,\infty)}(\sigma_l)] = \sum_{i=1}^{\infty}(l+2i)p_i = 3l$. Finally, $E[C_{D(\infty,\infty)}(\sigma_l)] = \sum_{i=1}^{\infty} 3ip_i = 3l$. Thus, in any case, the expected online cost is at least $3l$. The expected cost incurred by the adversary remains to be analyzed. If the input consists of at most l triples, the adversary acknowledges the packets individually; otherwise, it incurs a delay of $2l$ and acknowledges the packets of each triple together. Thus

$$\begin{aligned}E[C_{ADV}(\sigma_l)] &= \sum_{i=1}^{l} 3ip_i + \sum_{i=l+1}^{\infty} p_i(i+2l)\\ &= l + 2\sum_{i=1}^{l} ip_i + 2\sum_{i=l+1}^{\infty} p_i l\\ &= l + 2q\tfrac{1-(l+1)(1-q)^l+l(1-q)^{l+1}}{q^2} + 2l(1-q)^l\\ &= 3l - 4l(1-1/l)^l + 2l(1-1/l)^l.\end{aligned}$$

Thus $\lim_{l\to\infty} E[C_{ADV}(\sigma_l)]/l = 3 - 2/e$ and the theorem follows. □

THEOREM 4.3. *For the dynamic TCP acknowledgment problem with objective function f_p, no randomized online algorithm can achieve a competitive ratio smaller than $c \geq 2/(2-\frac{1}{e})$ against any oblivious adversary.*

Proof. An input σ_l, for any integer $l \geq 1$, consists of *pairs*. A pair is two packets that are $\sqrt[p]{l}$ time units apart. With probability $p_i = q(1-q)^{i-1}$, $\underline{q = 1/l}$, the input consists of i pairs for any $i \geq 1$. Pairs i and $i+1$ are separated by $\sqrt[p]{2l/p_{i+1}}$ time units. If a deterministic online algorithm acknowledges packets of different intervals together and if this happens for the first time for packets from pairs i and $i+1$, then the expected cost is at least $p_{i+1}(\sqrt[p]{2l/p_{i+1}})^p = 2l$. In the following we consider algorithms that never acknowledge packets from different pairs together, and we denote by $D(l')$, $l' \geq 1$, the algorithm that acknowledges packets in the first l' pairs separately and the packets in the $(l'+1)$st pair together. $D(\infty)$ is the algorithm that never acknowledges packets together. We have $E[C_{D(\infty)}(\sigma_l)] \geq \sum_{i=1}^{\infty} 2ip_i = 2q/q^2 = 2l$. For any $l \geq 0$,

$$\begin{aligned}E[C_{D(l')}(\sigma_l)] &\geq \sum_{i=1}^{l'} 2ip_i + \sum_{i=l'+1}^{\infty} (l'+i+l)p_i\\ &= E[C_{D(l'+1)}(\sigma_l)] - 2(l'+1)p_{l'+1} + (2l'+1+l)p_{l'+1} - \sum_{i=l'+2}^{\infty} p_i\\ &= E[C_{D(l'+1)}(\sigma_l)],\end{aligned}$$

and hence $E[C_{D(0)}(\sigma_l)] = [C_{D(l')}(\sigma_l)]$ for any $l' > 0$. We have $E[C_{D(0)}(\sigma_l)] = \sum_{i=1}^{\infty}(l+i)p_i = 2l$ and conclude that the expected online cost is at least $2l$.

The adversary acknowledges the packets of pairs separately if at most l intervals are sent; otherwise, it always acknowledges the packets of pairs together. Hence

$$\begin{aligned}E[C_{ADV}(\sigma_l)] &= \sum_{i=1}^{l} 2ip_i + \sum_{i=l+1}^{\infty} p_i(i+l)\\ &= l + \sum_{i=1}^{l} ip_i + \sum_{i=l+1}^{\infty} p_i l\\ &= l + q\tfrac{1-(l+1)(1-q)^l+l(1-q)^{l+1}}{q^2} + lq^l\\ &= 2l - 2l(1-1/l)^l + l(1-1/l)^l\end{aligned}$$

and $\lim_{l\to\infty} E[C_{ADV}(\sigma_l)]/l = 2 - 1/e$. The theorem holds. □

5. Conclusion and open problems. In this paper we have studied a TCP acknowledgment problem using objective functions that aim to keep the maximum acknowledgment delay of a data packet as short as possible. We presented tight upper and lower bounds on the performance of deterministic online algorithms. For randomized strategies we gave lower bounds. An interesting open problem is to devise tight bounds on the performance of randomized online algorithms. Additionally, it would be interesting to study TCP acknowledgment with objective functions that take into account the current network congestion. If the current congestion is low, it does not hurt to send many acknowledgments. On the other hand, if the congestion is high, it is worthwhile to send only few acknowledgments. Furthermore, in practice, the acknowledgments received by the sender affect the frequency by which packets are sent. It would be very interesting to incorporate such issues into the theoretical model.

REFERENCES

[1] A. Borodin and R. El-Yaniv, *Online Computation and Competitive Analysis*, Cambridge University Press, New York, 1998.

[2] D. R. Dooly, S. A. Goldman, and S. D. Scott, *TCP dynamic acknowledgement delay: Theory and practice*, in Proceedings of the 30th Annual ACM Symposium on Theory of Computing, ACM, New York, 1998, pp. 389–398.

[3] D. R. Dooly, S. A. Goldman, and S. D. Scott, *On-line analysis of the TCP acknowledgment delay problem*, J. ACM, 48 (2001), pp. 243–273.

[4] A. R. Karlin, C. Kenyon, and D. Randall, *Dynamic TCP acknowledgment and other stories about* $e/(e-1)$, in Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, ACM, New York, 2001, pp. 502–509.

[5] J. Noga, *private communication*, 2001.

[6] J. Noga, S. S. Seiden, and G. J. Woeginger, *A faster off-line algorithm for the TCP acknowledgment problem*, Inform. Process. Lett., 81 (2002), pp. 71–73.

[7] S. S. Seiden, *A guessing game and randomized online algorithms*, in Proceedings of the 32nd Annual ACM Symposium on Theory of Computing, ACM, New York, 2000, pp. 592–601.

[8] D. D. Sleator and R. E. Tarjan, *Amortized efficiency of list update and paging rules*, Commun. ACM, 28 (1985), pp. 202–208.

[9] A. C.-C. Yao, *Probabilistic computations: Towards a unified measure of complexity*, in Proceedings of the 18th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Los Alamitos, CA, 1977, pp. 222–227.

SIAM J. DISCRETE MATH.
Vol. 19, No. 4, pp. 952–965

WELL-COVERED VECTOR SPACES OF GRAPHS*

J. I. BROWN[†] AND R. J. NOWAKOWSKI[†]

Abstract. For any field $\mathbf{F}$, the set of all functions $f : V(G) \to \mathbf{F}$ whose sum on each maximal independent set is constant forms a vector space over $\mathbf{F}$. In this paper, we show that the dimension can vary depending on the characteristic of the field. We also investigate the dimensions of these vector spaces and show that while some families, such as chordal graphs, have unbounded dimension, other families, such as nonempty circulant graphs of prime order, have bounded dimension.

Key words. well-covered, independent, weighting, vector space, dimension

AMS subject classifications. 05C50, 05C69, 05C75

DOI. 10.1137/S0895480101393039

1. Introduction.

A *weighting* of a graph G is a function $f : V(G) \to \mathbf{F}$ that assigns a value from the field $\mathbf{F}$ to each vertex of G. Following [1], a *well-covered weighting* f of a graph G is a weighting such that $\sum_{x\in M} f(x)$ is constant for every maximal independent set M of G. For a well-covered weighting, we denote the common weight of the maximal independent sets as $f(G)$. In [1], the following is noted.

Observation 1. The well-covered weightings of a graph form a vector space.

This is clear since if f and g are well-covered weightings and k and l are elements of the field $\mathbf{F}$, then $kf + lg$ is also a well-covered weighting.

We remark that a *well-covered graph* [8] is a graph in which all maximal independent sets have the same cardinality. Thus, well-covered graphs G are precisely those graphs G for which $\mathbf{1}_G : V(G) \to \mathbf{F} : v \mapsto 1$ is a well-covered weighting over any field $\mathbf{F}$ of characteristic 0. The definition of the well-covered space can be traced to Caro and Yuster [2] in a more general setting. Let $H = (V, E)$ be a hypergraph and $\mathbf{F}$ be a field. A function $f : V \to \mathbf{F}$ is called *stable* if for each $e \in E$, the sum of the values of f on the members of e is the same. The stable functions form a vector space. One instance that Caro and Yuster consider is the space of well-covered weightings for a graph G. They denote this by $U(MIS : G, \mathbf{F})$ and the dimension by $\text{udim}(MIS : G, \mathbf{F})$ (MIS stands for maximal independent sets). In this paper we restrict ourselves to just well-covered weightings so we use $WC(G, \mathbf{F})$ and $\text{wcdim}(G, \mathbf{F})$ (we call the former the *well-covered space* of G and the latter the *well-covered dimension* of G). If the field has characteristic 0, then we eliminate the reference to $\mathbf{F}$ as well.

In general, our graph theoretic notation follows [3]. The complement of graph G is denoted by $\overline{G}$. The disjoint union of graphs G and H is denoted by $G \cup H$, and the join of G and H (which is $\overline{\overline{G} \cup \overline{H}}$) is denoted by $G + H$. A *maximum independent set* is one of maximum size (which is $\beta(G)$, the *independence number* of G). A *clique* is a complete subgraph (not necessarily maximal). We often obscure the difference between a subset of vertices of a graph and the subgraph they induce. Finally, for a vertex v of G, $N(v) = \{u \in V(G) : uv \text{ is an edge of } G\}$ is the *neighborhood* of v and

*Received by the editors July 26, 2001; accepted for publication (in revised form) June 27, 2005; published electronically January 6, 2006. This work was partially supported by a grant from the NSERC.

http://www.siam.org/journals/sidma/19-4/39303.html

[†]Department of Mathematics and Statistics, Dalhousie University, Halifax, NS, B3S 1E2, Canada (brown@mscs.dal.ca, rjn@mscs.dal.ca).

$N[v] = \{v\} \cup N(v)$ is the *closed neighborhood* of G. For matrix theoretic notation, we follow [7]. We denote the all ones vector of length n by $\mathbf{1}_n$, or simply $\mathbf{1}$ if the length is understood, and similarly use $\mathbf{0}_n$ to denote the all zeros vector of length n. (Vectors throughout are written as column vectors.)

If $I_1, \ldots, I_{t+1}$ are the maximal independent sets of G, then well-covered weightings are precisely the solutions to the *associated linear system*

$$\begin{aligned} \sum_{v \in I_1} x_v &= \sum_{v \in I_{t+1}} x_v, \\ \sum_{v \in I_2} x_v &= \sum_{v \in I_{t+1}} x_v, \\ &\cdots \\ \sum_{v \in I_t} x_v &= \sum_{v \in I_{t+1}} x_v \end{aligned}$$

(we call I_{t+1} the *common* maximal independent set for the linear system). This homogenous linear system can be written in matrix form as

$$A_G\,\mathbf{x} = \mathbf{0}$$

(we call the $t \times n$ matrix A_G an *associated matrix* for the graph G). Note that $\mathrm{wcdim}(G)$ equals the nullity of A_G (over $\mathbf{F}$) and hence is equal to the $|V(G)| - \mathrm{rank}(A_G)$ (where, of course, the rank is taken over $\mathbf{F}$). This formulation clearly shows that $\mathrm{wcdim}(G)$ depends only on the characteristic of $\mathbf{F}$, rather than the whole field.

As an illustration, consider W_5, the 5-wheel, which consists of a 5-cycle with a central vertex joined to each vertex on the 5-cycle. It is easy to discover (see Lemma 9) that all the vertices on the 5-cycle must have the same weight, and it is also easy to see that the central vertex must have weight equal to the sum of the weights of any maximal independent set of the 5-cycle, that is, twice the weight assigned to each vertex of the 5-cycle. Thus (writing the well-covered weightings as 6-tuples), we see that $WC(W_5, \mathbf{F})$ is spanned by $(1, 1, 1, 1, 1, 2)$ and hence has well-covered dimension 1. This example also shows that a basis for $WC(G, \mathbf{F})$ cannot always be chosen with values in $\{-1, 0, 1\}$ (when $\mathrm{char}(\mathbf{F}) \neq 2,\ 3$). As another example, we derive an upper bound on the well-covered dimension involving the chromatic number $\chi(G)$ of a graph G.

THEOREM 2. *Let G be a graph of order n. Then* $\mathrm{wcdim}(G) \leq n - \chi(G) + 1$.

Proof. For a graph G, let $\{I_i | i = 1, 2, \ldots, k\}$ be a sequence of nonempty, independent sets such that I_1 is a maximal independent set of G and for $j > 1$, I_j is a maximal independent set in $G - \cup_{i=1}^{j-1} I_i$. We extend each I_i to a maximal independent set I_i' of G. If we choose one vertex $v_i \in I_i$ for each $i = 1, \ldots, k$ of G, then using $I_1 = I_1'$ as the common maximal independent set for the linear system, the submatrix of A_G with rows corresponding to $I_2', \ldots, I_k'$ and columns corresponding to $v_2, \ldots, v_k$ is lower triangular with ones on the diagonal, as no v_i can lie in I_j' for $j < i$ (and in particular no v_i lies in I_1' for any $i = 2, \ldots, k$). Thus the rank of A_G is at least $k - 1$, so the nullity of A_G (and hence $\mathrm{wcdim}(G)$) is at most $n - k + 1$. Because $I_1, \ldots, I_k$ is a covering of G with k independent sets, $\chi(G) \leq k$, so $\mathrm{wcdim}(G) \leq n - k + 1 \leq n - \chi(G) + 1$. □

The major result on well-covered spaces can be found in Theorem 3.5 of [2]. There, it is shown that if the characteristic of $\mathbf{F}$ is 0, then for a connected graph $G \ncong C_7$ of girth 7 or greater, $\mathrm{wcdim}(G, \mathbf{F})$ equals the number of leaves. Moreover, the basis vectors can be taken to be the set $\{f_v | v \text{ is a leaf}\}$, where $f_v(v) = f_v(x) = 1$, x is the unique vertex adjacent to v (x is referred to as a *stem*), and $f_v(w) = 0$ otherwise. The exceptional case is $G \cong C_7$ in which case $\mathrm{wcdim}(G, \mathbf{F}) = 1$ and the basis vector is the all ones vector. All bases can be constructed in polynomial time, and the restriction on the field can be removed if there is at least one leaf. In particular, Caro and Yuster's result shows that the well-covered dimension of a tree is equal to the number of leaves.

In this paper, after illustrating how the well-covered dimension can depend on the characteristic of the field, we restrict ourselves to the most interesting case, characteristic 0, and consider families of graphs for which the well-covered dimension is unbounded and those for which it is bounded. Extending Caro and Yuster's result that the well-covered dimension of a tree is equal to the number of leaves, we calculate the dimension of chordal graphs and show how a corresponding basis can be derived from the chordal graph's simplicial decomposition. Using linear algebraic techniques, we show on the other hand that nonempty circulant graphs of prime order have bounded dimension over any field of characteristic 0.

2. Characteristic does make a difference. In this section we provide, for every prime p, an infinite number of graphs whose dimension is different over fields of characteristic p and 0.

We begin by defining graphs $G_{p,q,n}$. Let $n \equiv 0 \bmod p$ with $n > p \geq 3$ (we will handle the case $p = 2$ at the end). Let $q > p(p-1)$, $q \not\equiv 0 \bmod p$. We form $G_{p,q,n}$ on vertex sets $V_0, \ldots, V_{q-1}$, where $V_i = \{v_{i,1}, \ldots, v_{i,n}\}$. The *nonedges* of $G_{p,q,n}$ are $v_{i,r}v_{i,s}$ and $v_{i,r}v_{j,r}$, with $r,\ s = 1, 2, \ldots, n$, $r \neq s$, $i,\ j \in \{0, 1, \ldots, q-1\}$, $i - j \in \{1, 2, \ldots, p-1\}$ (arithmetic mod q). The complement of $G_{3,7,6}$ (which has fewer edges than $G_{3,7,6}$) is shown in Figure 1. Now it is not difficult to verify that the maximal independent sets of $G_{p,q,n}$ are $V_0, \ldots, V_{q-1}$ together with the sets

$$\{v_{i,k}, v_{i+1,k}, \ldots, v_{i+p-1,k}\}$$

(here and elsewhere, addition is modulo q). Setting the sum of each of the weights on the maximal independent sets equal to the sum of the weights on the vertices of V_{q-1}, we find that the linear system corresponding to the well-covered weightings is $A\mathbf{x} = \mathbf{0}$, where

$$A = \left(\begin{array}{ccccccccc}
I_n & I_n & \cdots & I_n & 0_n & 0_n & \cdots & 0_n & -J_n \\
0_n & I_n & \cdots & I_n & I_n & 0_n & \cdots & 0_n & -J_n \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0_n & 0_n & \cdots & I_n & I_n & I_n & \cdots & I_n & -J_n \\
0_n & 0_n & \cdots & 0_n & I_n & I_n & \cdots & I_n & I_n - J_n \\
I_n & 0_n & \cdots & 0_n & 0_n & I_n & \cdots & I_n & I_n - J_n \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
I_n & I_n & \cdots & I_n & 0_n & 0_n & \cdots & 0_n & I_n - J_n \\
\mathbf{1}_n^T & \mathbf{0}_n^T & \cdots & \mathbf{0}_n^T & \mathbf{0}_n^T & \mathbf{0}_n^T & \cdots & \mathbf{0}_n^T & -\mathbf{1}_n^T \\
\mathbf{0}_n^T & \mathbf{1}_n^T & \cdots & \mathbf{0}_n^T & \mathbf{0}_n^T & \mathbf{0}_n^T & \cdots & \mathbf{0}_n^T & -\mathbf{1}_n^T \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\mathbf{0}_n^T & \mathbf{0}_n^T & \cdots & \mathbf{0}_n^T & \mathbf{0}_n^T & \mathbf{0}_n^T & \cdots & \mathbf{1}_n^T & -\mathbf{1}_n^T
\end{array} \right),$$

the columns are indexed by the vertices

$$v_{0,1}, v_{0,2}, v_{0,3}, \ldots, v_{i,1}, v_{i,2}, v_{i,3}, \ldots, v_{q-1,n},$$

and the rows are indexed by the maximal independent sets $V_0, V_1, V_2, \ldots, V_{q-1}$ and

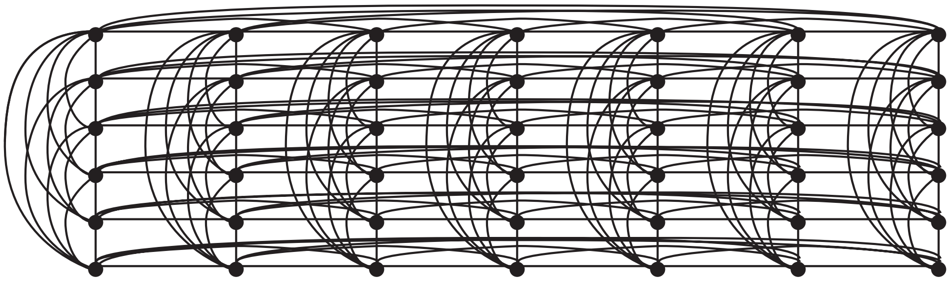


FIG. 1. $\overline{G_{3,7,6}}$.

the sets

$$\{v_{0,1}, v_{0+1,1}, \ldots, v_{0+p-1,1}\}, \{v_{0,2}, v_{0+1,2}, \ldots, v_{0+p-1,2}\}, \ldots, \{v_{1,1}, v_{1+1,1}, \ldots, v_{1+p-1,1}\},$$
$$\ldots, \{v_{q-1,1}, v_{q-1+1,1}, \ldots, v_{q-1+p-1,1}\}, \ldots \{v_{q-1,1}, v_{q-1+1,1}, \ldots, v_{q-1+p-1,1}\}.$$

In the above block form of the matrix, the subscript n denotes the order of the submatrix, with J_n being the $n \times n$ matrix of all ones and 0_n being the $n \times n$ matrix of all zeros. If B denotes the top nq rows of A, then $B = C - D$, where

$$C = \begin{pmatrix} I_n & I_n & \cdots & I_n & 0_n & 0_n & \cdots & 0_n & 0_n \\ 0_n & I_n & \cdots & I_n & I_n & 0_n & \cdots & 0_n & 0_n \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0_n & 0_n & \cdots & I_n & I_n & I_n & \cdots & I_n & 0_n \\ 0_n & 0_n & \cdots & 0_n & I_n & I_n & \cdots & I_n & I_n \\ I_n & 0_n & \cdots & 0_n & 0_n & I_n & \cdots & I_n & I_n \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ I_n & I_n & \cdots & I_n & 0_n & 0_n & \cdots & 0_n & I_n \end{pmatrix}$$

is block circulant (with p consecutive identity matrices in each block row) and

$$D = \begin{pmatrix} 0_n & 0_n & \cdots & 0_n & 0_n & 0_n & \cdots & 0_n & J_n \\ 0_n & 0_n & \cdots & 0_n & 0_n & 0_n & \cdots & 0_n & J_n \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0_n & 0_n & \cdots & 0_n & 0_n & 0_n & \cdots & 0_n & J_n \\ 0_n & 0_n & \cdots & 0_n & 0_n & 0_n & \cdots & 0_n & J_n \\ 0_n & 0_n & \cdots & 0_n & 0_n & 0_n & \cdots & 0_n & J_n \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0_n & 0_n & \cdots & 0_n & 0_n & 0_n & \cdots & 0_n & J_n \end{pmatrix}.$$

Suppose first that the characteristic of $\mathbf{F}$ is 0. Then by summing the first n rows and subtracting off rows $nq + 1$ to $nq + p$, we get a row with $n(q - 1)$ zeros followed by $-(n - p)\mathbf{1}_n^T$. Since $n > p$ and the characteristic is 0, we can divide through by $-(n - p)$ to get $\mathbf{1}_n^T$ in the last positions. Adding this row to each of the first nq rows of A, we obtain a matrix whose upper nq rows are the block circulant C. It is clear that C is nonsingular iff the $q \times q$ circulant matrix formed by replacing each I_n and 0_n by 1 and 0, respectively, is nonsingular.

However, it is known (cf. [7, p. 66]) that the determinant of a circulant with first row $a_1, \ldots, a_m$ is

$$\prod \sum_{i=1}^{m} a_i x^{i-1},$$

where the product is taken over all mth roots x of unity. In our case, the determinant of C is given by

$$\prod \sum_{i=1}^{p} x^{i-1}$$

over all x that are qth roots of unity. However, no term in this product is 0, since clearly the term with $x = 1$ is nonzero, and for any other qth root of unity x we have (by multiplying through by $1 - x$) that x is also a pth root of unity, a contradiction since $q \not\equiv 0 \bmod p$. Thus we conclude that the matrix A has full row rank over the field of characteristic 0 and hence has nullity 0, i.e., $\mathrm{wcdim}(G_{p,q,n}, \mathbb{Q}) = 0$.

On the other hand, if we weight every vertex with 1, then this yields a weighting over a field of characteristic p since the maximal independent sets have weight p or weight $n \equiv 0 \bmod p$. Thus $\mathrm{wcdim}(G_{p,q,n}, \mathbb{Z}_p) > 0$.

Last, we handle $p = 2$. For any $n > 2$, n even, we form the graph $G_{2,n}$ by removing a perfect matching from $K_{n,n}$. We let the partition be $V_0 = \{a_1, \dots, a_n\}$ and $V_1 = \{b_1, \dots, b_n\}$, with $a_1b_1, \dots, a_nb_n$ being the perfect matching that is removed. The maximal independent sets are $\{a_i, b_i\}$ for $i = 1, 2, \dots, n$ and V_1 and V_2. Setting the sum of each of the weights on the maximal independent sets equal to that of the weights on the vertices of V_2, we find that the linear system corresponding to the well-covered weightings is

$$A\mathbf{x} = \mathbf{0},$$

where

$$A = \begin{pmatrix} I_n & I_n - J_n \\ \mathbf{1}_n^T & -\mathbf{1}_n^T \end{pmatrix}.$$

Subtracting the top n rows from the bottom yields

$$\begin{pmatrix} I_n & I_n - J_n \\ \mathbf{0}_n^T & (n-2)\mathbf{1}_n^T \end{pmatrix}.$$

Over $\mathbb{Q}$ we can divide out by $n - 2$ so that A is row equivalent to

$$\begin{pmatrix} I_n & I_n - J_n \\ \mathbf{0}_n^T & \mathbf{1}_n^T \end{pmatrix} \qquad (*),$$

which has rank $n+1$. Hence the nullity is $n-1$, which implies that $\mathrm{wcdim}(G_{2,n}, \mathbb{Q}) = n - 1$.

On the other hand, over $\mathbb{Z}_2$, since n is even, A is row equivalent to

$$\begin{pmatrix} I_n & I_n - J_n \\ \mathbf{0}_n^T & \mathbf{0}_n^T \end{pmatrix},$$

which has rank n. Hence the nullity is n, which implies that $\mathrm{wcdim}(G_{2,n}, \mathbb{Z}_2) = n$.

For the remainder of the paper, we shall restrict our discussion to fields of characteristic 0, though some of the results will hold over fields of other characteristic as well.

3. Families of graphs with unbounded well-covered dimension. In this section, we shall determine (in polynomial time) the well-covered dimension of cographs and chordal graphs, where the latter extends the result of Caro and Yuster on trees. We begin with the easier case.

3.1. Cographs and anti-well-covered graphs. A *cograph* is a graph that does not contain an induced path on four vertices. It is well known (cf. [5]) that cographs have a recursive definition; the class of cographs is the smallest class of graphs containing K_1 (the complete graph on one vertex) that is closed under disjoint union and join. We shall need to introduce a definition that is of interest in its own right.

DEFINITION 3. *A graph for which $f(G) = 0$ for every well-covered weighting f of G is called an anti-well-covered graph.*

Note that in a well-covered graph G of order n, the all ones vector $\mathbf{1}_n$ is in $WC(G, \mathbf{F})$, and for an anti-well-covered graph, $\mathbf{1}_n$ is in $WC(G, \mathbf{F})^{\perp}$, the orthogonal complement of the well-covered space of G. The fact that $WC(G, \mathbf{F})^{\perp} \cap WC(G, \mathbf{F}) = \{\mathbf{0}\}$ ensures that no well-covered graph is anti-well-covered, and this motivates our choice of name for the property.

A graph of dimension 0 is clearly an anti-well-covered graph, but there are others. For example, one can verify that C_6 and Q_3 (the 3-cube) are anti-well-covered. Also, $K_{n,n} - M$, where $n > 2$ and M is a 1-factor, is an anti-well-covered graph with dimension n over any field of characteristic c, where $\gcd(n, c) = 1$ (this follows from the derivation of $(*)$ in the previous section). In order to determine the well-covered dimension of cographs, we will need some simple properties of anti-well-covered graphs.

LEMMA 4. *Let G or H be graphs. Then $G \cup H$ is anti-well-covered iff both G and H are anti-well-covered, whereas $G + H$ is anti-well-covered iff either G or H is anti-well-covered.*

Proof. The well-covered weightings of the disjoint union of two graphs G and H are precisely those functions on $V(G) \cup V(H)$ whose restrictions to G and H are well-covered weightings, whereas the well-covered weightings of the join of G and H are precisely those functions on $V(G) \cup V(H)$ whose restrictions to G and H are well-covered weightings *with the same sum.* It follows that $G \cup H$ is anti-well-covered iff both G and H are anti-well-covered, whereas $G + H$ is anti-well-covered iff either G or H is anti-well-covered. □

We now determine how the well-covered dimension behaves under disjoint union and join.

LEMMA 5. *Let G and H be graphs. Then*

1. $\text{wcdim}(G \cup H) = \text{wcdim}(G) + \text{wcdim}(H)$, *and*

2. $\text{wcdim}(G + H) = \text{wcdim}(G) + \text{wcdim}(H) - 1$ *unless both G and H are anti-well-covered graphs in which case* $\text{wcdim}(G + H) = \text{wcdim}(G) + \text{wcdim}(H)$.

Proof. The first result is given in [1]. Let L be the subspace generated by those vectors whose restrictions to G and H are well-covered weightings on the respective graphs. From the proof of Lemma 4, L properly contains the subspace generated by well-covered weightings of $G + H$ iff either G or H is anti-well-covered. (If say G is not anti-well-covered, then we can find well-covered weightings of G with weight equal to any field element, in particular, of unequal weight to some weighting of H.) Thus $\text{wcdim}(G + H) = \text{wcdim}(G) + \text{wcdim}(H)$ if both G and H are anti-well-covered, and $\text{wcdim}(G + H) < \text{wcdim}(G) + \text{wcdim}(H)$ otherwise. In the latter case, note that if we write corresponding linear systems defining the subspaces of G and H as

$$A_G\, \mathbf{x} = \mathbf{0}$$

and

$$A_H\, \mathbf{x} = \mathbf{0},$$

then a corresponding linear system for $G+H$ can be given as

$$A_{G+H}\,\mathbf{x}=\mathbf{0},$$

where

$$A_{G+H}=\begin{pmatrix} A_G & 0\\ 0 & A_H\\ \mathbf{u} & \mathbf{v}\end{pmatrix}$$

with $\mathbf{u}$ and $\mathbf{v}$ being nonzero vectors of the appropriate dimension. Now

$$\operatorname{rank}(A_{G+H})\le \operatorname{rank}(A_G)+\operatorname{rank}(A_H)+1,$$

so since $\operatorname{wcdim}(K)=\operatorname{nullity}(A_K)=|V(K)|-\operatorname{rank}(A_K)$ for any graph K, we find that

$$\operatorname{wcdim}(G+H)\ge \operatorname{wcdim}(G)+\operatorname{wcdim}(H)-1.$$

Since $\operatorname{wcdim}(G+H)<\operatorname{wcdim}(G)+\operatorname{wcdim}(H)$, we conclude that $\operatorname{wcdim}(G+H)=\operatorname{wcdim}(G)+\operatorname{wcdim}(H)-1$. □

THEOREM 6. *The dimension of a cograph can be determined in polynomial time.*

Proof. A cograph is constructed via the disjoint union and join operation from K_1. A cograph can be recognized and the order of operations for its construction can be determined in polynomial time [5]. It follows that we can recognize whether a cograph is anti-well-covered in polynomial time as well. The dimension can be determined in polynomial time from Lemma 4. □

We conclude this section by applying anti-well-covered graphs to determining the dimension of graphs with independence number 2. Graphs with independence number 1 are complete, and it is easy to see that these all have dimension 1, with all vertices having the same weight in any weighting.

THEOREM 7. *Let G be a graph with $\beta(G)=2$. Then* $\operatorname{wcdim}(G)$ *is* 1 *plus the number of bipartite components of order of at least* 2 *in the complement* $\overline{G}$ *of* G.

Proof. Let the components of $\overline{G}$ be $D_1,\dots,D_t$. Noting that K_1 is not anti-well-covered, we observe from Lemma 5 that any D_i of order 1 does not affect the dimension, so we can assume that each D_i has an order of at least 2. Also, every edge of $\overline{G}$ is a maximal independent set of G. Note that under any weighting of G, if xy and yz are edges of $\overline{G}$, then x and z have equal weight, so that any two vertices connected by a walk of even length have the same weight.

Consider any component D of $\overline{G}$. If D is not bipartite, it contains an odd cycle. By the argument above (traveling twice around the cycle), any well-covered weighting must be constant on this cycle, and indeed on the component D, and hence the subgraph of G induced by D has dimension 1.

On the other hand, if D is bipartite with bipartition (X,Y), then we can weight every vertex of X with one weight, weight every vertex of Y with another, and derive a well-covered weighting of the graph. Moreover, every well-covered weighting of G necessarily assigns the same weights to vertices of X and the same weights to the vertices of Y, as vertices of X are at even distances from one another (similarly for the vertices of Y). Thus the subgraph of G induced by D has dimension 2.

Now each D_i induces a well-covered graph (with $\beta=2$), so in particular, no D_i is anti-well-covered. Since G is the join of the subgraphs induced by $D_1,\dots,D_t$, we conclude the stated formula for $\operatorname{wcdim}(G)$ from Lemma 5. □

3.2. Complements of k-trees, chordal graphs, and related vertices. In this section we show that certain other well known families of graphs also have unbounded well-covered dimension. A *k-tree,* ($k \geq 2$) is defined recursively: G_0 is a k-clique; for $i > 0$, G_i is formed from G_{i-1} by adding a new vertex that is joined to a $(k-1)$-clique of G_{i-1}. Every tree is a 2-tree. Here we determine the well-covered dimension of *complements* of k-trees. The dimension of k-trees themselves will be covered later in this section.

Theorem 8. *If G is the complement of a k-tree, then G has dimension k.*

Proof. Let G be the complement of a k-tree with G_0 an independent set of size k of G and $G_1, G_2, \ldots, G_m \cong \overline{G}$ a sequence of k-trees that build to $\overline{G}$. Let the vertices of G_0 be $v_1, \ldots, v_k$. Let f be any well-covered weighting of G. By induction on i we show that (i) the maximal independent sets of $\overline{G_i}$ are the independent sets of size k of $\overline{G_i}$ and (ii) $\mathrm{wcdim}(\overline{G_i}) = k$. The latter, for $i = m$, completes the proof.

For $i = 0$, (i) and (ii) are obvious. Suppose now that G_i is formed from G_{i-1} by the addition of vertex v_{k+i} so that, for some independent set X_i of size $k-1$ of $\overline{G_{i-1}}$, v_{k+1} is joined to all of $\overline{G_{i-1}} - X_i$ but no vertex of X_i. Now the maximal independent sets of $\overline{G_i}$ are those that do not contain v_{k+i} (which are the maximal independent sets of $\overline{G_{i-1}}$) and those that contain v_i, of which there is only one, namely $\{v_i\} \cup X_i$. Thus by induction (i) holds. Moreover, an associated linear system for $\overline{G_i}$ can be derived from that of $\overline{G_{i-1}}$ by adding in the equation

$$\sum_{v \in \{v_i\} \cup X_i} x_v = \sum_{v \in G_0} x_v.$$

This introduces a new variable, so it is not hard to see that the associated matrices A_{i-1} and A_i have the same nullity (since A_{i+1} has a rank one larger than that of A_i, but one more column). Part (ii) now follows. □

We now turn our attention to chordal (or *triangulated*) graphs, that is, graphs without an induced cycle of length of at least 4. Every chordal graph has a *simplicial decomposition*; that is, the graph can be recursively built from a complete graph by adding vertices that are joined to cliques (for more information on chordal graphs, cf. [6, p. 83]). Note that all trees and all k-trees are chordal graphs. We now calculate the dimension of chordal graphs. A new relation on the vertices of a graph plays a key role in calculating the well-covered dimension of chordal graphs. Two vertices x and y of a graph are *related* if there is an independent set I, containing neither x nor y, such that $I \cup \{x\}$ and $I \cup \{y\}$ are both maximal independent sets. Note that x and y must be adjacent or else both could be added to I.

Lemma 9. *Let f be a well-covered weighting of G. If x and y are related vertices in G, then $f(x) = f(y)$.*

Proof. For an appropriate independent set I, $f(x)+\sum_{z\in I} f(z) = f(y)+\sum_{z\in I} f(z)$, and the result follows. □

Now we say a vertex x of a graph G is *simplicial* if $N[x]$ is a maximal clique. Let $\mathcal{C}(G) = \{C | C \text{ is a maximal clique containing a simplicial vertex of } G\}$. The members of $\mathcal{C}(G)$ are called *simplicial cliques.* Let $\mathrm{sc}(G) = |\mathcal{C}(G)|$. Let C be a simplicial clique of G, and let f_C be the *associated weighting*: $f_C(v) = 1$ if $v \in C$ and $f_C(v) = 0$ otherwise. It was shown in [2] that the number of leaves of a graph is a lower bound to its dimension. We generalize this to simplicial cliques.

Lemma 10. *Let G be a graph. Then $\{f_C | C \in \mathcal{C}\}$ is an independent set of vectors and* $\mathrm{wcdim}(G) \geq \mathrm{sc}(G)$.

Proof. Let $C \in \mathcal{C}$. There is a vertex $v \in C$ that is adjacent only to vertices of C. Therefore, any maximal independent set must contain exactly one vertex of C, and so f_C is a well-covered weighting. Moreover, v is in no other maximal simplicial clique. Therefore, $f_C(v) = 1$, but $f_D(v) = 0$ for all $D \in \mathcal{C}$, $D \neq C$. Consequently $\{f_C | C \in \mathcal{C}\}$ is an independent set of well-covered weightings. The second part of the lemma now follows. □

Our main result proves that equality indeed holds in Lemma 10 for chordal graphs.

THEOREM 11. *Let G be a chordal graph. Then* $\mathrm{wcdim}(G) = \mathrm{sc}(G)$.

The remainder of the section is devoted to a proof of Theorem 11.

From Lemma 10 we have $\mathrm{wcdim}(G) \geq \mathrm{sc}(G)$. The second part of the proof is now by induction on the size of G. If G is a singleton, then $\mathrm{wcdim}(G) = \mathrm{sc}(G) = 1$. Assume that the result is true for all chordal graphs of sizes 1 through k for some $k \geq 1$. We shall need a few observations about simplicial cliques.

Observation 12. Let w and y be adjacent vertices. If w is a simplicial vertex, then $N[w] \subseteq N[y]$. If both w and y are simplicial vertices, then $N[w] = N[y]$, so that both w and y "generate" the same simplicial clique of G. □

Let x be a simplicial vertex of G, and put $H = G - \{x\}$. Note that H is also chordal. By induction, $\mathrm{sc}(H) = \mathrm{wcdim}(H)$.

Observation 13. Consider a simplicial clique $C \in \mathcal{C}(H)$. If there is a simplicial vertex $y \in C$ and y is not adjacent to x, then $C \in \mathcal{C}(G)$. Similarly, if $D \in \mathcal{C}(G)$ and there is a simplicial vertex $z \in D$, $z \neq x$, with z not adjacent to x, then $D \in \mathcal{C}(H)$. □

Observation 14. If $C \in (\mathcal{C}(G) - \mathcal{C}(H))$, then $C = N[x]$.

Proof. By Observation 13, all simplicial vertices of C are adjacent to x, but then, by Observation 12, we have $C = N[x]$. □

Observation 15. If $C \in (\mathcal{C}(H) - \mathcal{C}(G))$, then either $C = N(x)$, or there is a simplicial vertex $y \in C$, y adjacent to x. Moreover, there is at most one such simplicial clique C.

Proof. Suppose that $C \in (\mathcal{C}(H) - \mathcal{C}(G))$, and let $y \in C$ be a simplicial vertex in H. It follows from Observation 13 that y is adjacent to x (else $C \in \mathcal{C}(G)$) so that in G we have, by Observation 12, $N[x] \subseteq N[y]$. If y is a simplicial vertex of G, then by Observation 12 $N[x] = N[y]$ and thus $C = N[x] - x = N(x)$. If y is not a simplicial vertex of G, then, in H, $C = N[y] = N(x) \cup A$. Suppose that $C, D \in (\mathcal{C}(H) - \mathcal{C}(G))$ with $C \neq D$. There are simplicial vertices $y \in D$, y adjacent to x, and $z \in C$ which is also adjacent to x. But then z and y are adjacent (since both are in the clique $N(x)$), and so by Observation 12, $C = N[z] = N[y] = D$. Thus, there is at most one simplicial clique $C \in (\mathcal{C}(H) - \mathcal{C}(G))$. □

Observation 16. $\mathrm{sc}(G) - 1 \leq \mathrm{sc}(H) \leq \mathrm{sc}(G)$.

Proof. By Observation 13, every simplicial clique of H that does not contain a simplicial vertex from $N(x)$ is a simplicial clique of G, and by Observation 15 there is at most one simplicial clique of H with a vertex in $N(x)$. Since G has $N[x]$ as a simplicial clique while H clearly does not, we have $\mathrm{sc}(H) \leq \mathrm{sc}(G)$. On the other hand, there is only one simplicial clique of G, namely $N[x]$, that is not a simplicial clique of H, as the only other simplicial vertices of G in $N[x]$ generate the same simplicial clique (by Observation 12). Thus $\mathrm{sc}(G) - 1 \leq \mathrm{sc}(H)$. □

Now back to the proof of Theorem 11. Let $f(G)$ be a well-covered weighting of G, and let K be the (common) sum of the weights of a maximal independent set. We first show that any well-covered weighting of G can be associated with a well-covered weighting of H. We then use this and the fact that $\mathrm{wcdim}(H) = \mathrm{sc}(H)$ to show that $\text{wcdim}(G) = \text{sc}(G)$. From the observations we see that there are three cases to consider.

1. $\mathcal{C}(H) \subset \mathcal{C}(G)$, i.e., no new simplicial clique is created when x is deleted,
2. $\{C\} = \mathcal{C}(H) - \mathcal{C}(G)$ and $C = N(x)$, or
3. $\{C\} = \mathcal{C}(H) - \mathcal{C}(G)$ and $C \neq N(x)$.

Case 1. We have $\mathcal{C}(H) \subset \mathcal{C}(G)$ so that $\{N[x]\} = \mathcal{C}(G) - \mathcal{C}(H)$ and $\text{sc}(G) = \text{sc}(H) + 1$. Since every simplicial clique of H is a simplicial clique of G, then, from Observation 2, it follows that for all $y \in N(x)$, y is not simplicial in H. We define a weighting w_f on $V(H)$ by

$$w_f(v) = \begin{cases} f(v) & \text{if } v \text{ is not adjacent to } x, \\ f(v) - f(x) & \text{if } v \in N(x). \end{cases}$$

We claim that w_f is in fact a well-covered weighting of H. Let I be a maximal independent set of H. If there exists $s \in I$ such that $s \in N(x)$, then I is a maximal independent set in G, and moreover no other vertex in I is adjacent to x. Therefore,

$$\sum_{v \in I} w_f(v) = \sum_{v \in I-s} w_f(v) + w_f(s) = \sum_{v \in I-s} f(v) + f(s) - f(x) = K - f(x).$$

If I contains no vertex adjacent to x, then $I \cup \{x\}$ is a maximal independent set in G. Therefore

$$f(x) + \sum_{v \in I} w_f(v) = f(x) + \sum_{v \in I} f(v) = K;$$

i.e., $\sum_{v \in I} w_f(v) = K - f(x)$. Thus w_f is a well-covered weighting of H. In H, let h_i, $i = 1, 2, \ldots, \text{sc}(H)$ be the vector with weight 1 on the coordinates corresponding to the vertices of the ith simplicial clique. By induction, this is a basis for $\text{wcdim}(H)$. In G, we extend these vectors to g_i, $i = 1, 2, \ldots, \text{sc}(H)$, where g_i is the vector with weight 1 on the vertices of the coordinates corresponding to the ith simplicial clique. (That is, each g_i is the same as h_i, but a value for $g_i(x) = 0$ is now defined.) In this case, since every simplicial clique of H is a simplicial clique of G, by Lemma 10, the g_i's are linearly independent, well-covered weightings of G. Now w_f is a well-covered weighting of H, and so

$$w_f = \sum_{i=1}^{\text{sc}(H)} c_i h_i.$$

Now, by the construction of w_f, $f(v) - \sum_{i=1}^{\text{sc}(H)} c_i h_i(v) = 0$ for $v \notin N(x)$, and so the well-covered weighting $g = f - \sum_{i=1}^{\text{sc}(H)} c_i g_i$ is nonzero only on vertices of $N[x]$. For any $w \in N[x]$, extend w to a maximal independent set $I(w)$ of G. Then $\sum_{u \in I(w)} g(u) = g(w)$, but g is a well-covered weighting so that g is a constant on the simplicial clique $N[x]$ and 0 is everywhere else, i.e., g is a scalar multiple of the associated weighting of the simplicial clique $N[x]$ of G. Thus $f = g + \sum_{i=1}^{\text{sc}(H)} c_i g_i$ is a linear combination of the associated weightings for the simplicial cliques of G, and we conclude that $\text{wcdim}(G) \leq \text{sc}(G)$, and hence (by Lemma 10) $\text{wcdim}(G) = \text{sc}(G)$ in this case.

Case 2. We have $\{C\} = \mathcal{C}(H) - \mathcal{C}(G)$ and $C = N(x)$. Therefore, there is a $y \in C$ which is simplicial in both H and G. Let I be any maximal independent set of $G - N[x]$. Then both $I \cup \{x\}$ and $I \cup \{y\}$ are maximal independent sets for G, i.e., x and y are related and thus have the same weight in any well-covered weighting of G. Note that the restriction f' of f to H is also a well-covered weighting. This follows since any maximal independent set I of H must contain a vertex of $C = N(x)$, and thus I is also a maximal independent set of G. Let h_i, $i = 1, 2, \ldots, \mathrm{sc}(H)$, be the vector with weight 1 on the coordinates corresponding to the vertices of the ith simplicial clique of H, and let C correspond to $i = 1$. By induction, this is a basis for $WC(H)$; therefore, $f = \sum_{i=1}^{\mathrm{sc}(H)} d_i h_i$ and $d_1 h_1 = f' - \sum_{i=2}^{\mathrm{sc}(H)} d_i h_i$. In G, we extend these vectors to g_i, $i = 2, \ldots, \mathrm{sc}(H)$, with g_i the vector having weight 1 on the coordinates corresponding to the vertices of the ith ($i > 1$) simplicial clique of H (and G). By Lemma 10, each g_i is a well-covered weighting of G and thus so is $g = f - \sum_{i=2}^{\mathrm{sc}(G)} d_i g_i$. Under g, the only vertices with nonzero weights are those of $N[x]$. All of the vertices of C have the same weight under g since g restricted to C is h_1. But since $f(x) = f(y)$ (y simplicial in C), it follows that g is constant on $N[x]$ and that $\{g_i | i = 2, \ldots, \mathrm{sc}(H)\} \cup \{g\}$ spans $WC(G)$. In this case, again we have that $\mathrm{wcdim}(G) = \mathrm{sc}(G)$.

Case 3. We have $\{C\} = \mathcal{C}(H) - \mathcal{C}(G)$ and $C \neq N(x)$. Therefore, there is a simplicial vertex $y \in C$, y adjacent to x, y not simplicial in G, and $C = N[y] - \{x\} = N(x) \cup A$. Also, in H, if $z \in A$ were a simplicial vertex, then, by Observation 1, $N[y] - \{x\} = N[z]$ and $C = N[z]$ would also be a simplicial clique in G. Therefore A contains no simplicial vertices. We define a weight function w_f on $V(H)$ by

$$w_f(v) = \begin{cases} f(v) + f(x) & \text{if } v \in A, \\ f(v) & \text{otherwise.} \end{cases}$$

Let I be a maximal independent set of H. If there exists $s \in I$ such that $s \in N(x)$, then I is a maximal independent set in G. Thus

$$\sum_{v \in I} w_f(v) = \sum_{v \in I} w_f(v) = K.$$

If I contains no vertex adjacent to x, then it must contain exactly one vertex $z \in A$, and $I \cup \{x\}$ must be a maximal independent set in G. Therefore,

$$\begin{aligned} \sum_{v \in I} w_f(v) &= w_f(z) + \sum_{v \in I - \{z\}} w_f(v) \\ &= f(x) + f(z) + \sum_{v \in I - \{z\}} f(v) \\ &= K. \end{aligned}$$

Thus, w_f is a well-covered weighting of H.

In H, let h_i, $i = 1, 2, \ldots, \mathrm{sc}(H)$, be the vector with weight 1 on the vertices of the coordinates corresponding to the ith simplicial clique where the simplicial clique containing y has index 1. By induction, this is a basis for $WC(H)$. In G, let g_i, $i = 2, 3, \ldots, \mathrm{sc}(H)$, be the vector with weight 1 on the coordinates corresponding to the vertices of the ith simplicial clique. Recall that in this case we have $\mathrm{sc}(G) = \mathrm{sc}(H)$ and the simplicial cliques of H with indices 2 through $\mathrm{sc}(H)$ are also simplicial cliques in G. Thus, $\{g_i | i = 2, 3, \ldots, \mathrm{sc}(G)\}$ is a linearly independent set. Now w_f is a well-covered weighting of H, and so

$$w_f = \sum_{i=1}^{\mathrm{sc}(H)} c_i h_i.$$

Therefore,

$$w_f - \sum_{i=2}^{\mathrm{SC}(H)} c_i h_i = c_1 h_1,$$

i.e., all the vertices of $N[y] \cap H$ have weight c_1 in the well-covered weighting $w_f - \sum_{i=2}^{\mathrm{SC}(H)} c_i h_i$ of H. Therefore, in G, the only vertices with nonzero weight in the well-covered weighting $g = f - \sum_{i=2}^{\mathrm{SC}(G)} c_i g_i$ of G are the vertices of $N[y]$ with $g(z) = c_1 - f(x)$ for all $z \in A$ and $g(z) = c_1$ for $z \in N(x)$, and $g(x) = f(x)$.

We now need to show that $c_1 = f(x)$, and for that we need to find an independent set with certain properties. Let I be a minimum-sized independent set of $V(G) - (C \cup \{x\})$ that dominates (i.e., is adjacent to) the maximum number of vertices in C. If I does not dominate all the nonsimplicial vertices of C, then there exists a nonsimplicial $z \in C$ which is not dominated by a vertex of I. However, since z is not simplicial there exists $w \in G - (N[x] \cup N[y])$ with z adjacent to w. Now, since $I \cup \{w\}$ is not independent (I was maximum with this domination property), there exists $i \in I$ such that i is adjacent to w. Let $s \in C \cap N(i)$. The latter is nonempty since otherwise i could be deleted from I, a contradiction. Thus s is adjacent to z since C is a clique, and consequently, $\{z, w, i, s\}$ is a C_4. Since H is chordal, this cycle must have a chord, specifically $w \sim s$. Since this is true for any i and s, we can replace all the neighbors of w in I by w. This independent set dominates more vertices in C than does I, and this is a contradiction. Therefore, there is an independent set J of $V(G) - (C \cup \{x\})$ which dominates all the nonsimplicial vertices in C and in particular all of A (recall that A has no simplicial vertices). Now, since J dominates all of A, $J \cup \{x\}$ and $J \cup \{y\}$ are maximal independent sets, and so x and y are related and, in particular, $g(x) = g(y) = c_1$. Thus $g(x) = g(w)$ for any $w \in N(x)$. But then for all $z \in A$, $g(z) = c_1 - g(x) = 0$. It follows that the original well-covered weighting f is a linear combination of $\{g_i | i = 2, 3, \ldots, \mathrm{sc}(G)\} \cup \{g'\}$, where g' is 1 on the vertices of $N[x]$ and is 0 everywhere else. Thus, in this and all cases, $\mathrm{wcdim}(G) = \mathrm{sc}(G)$, and the theorem is proved. $\square$

We remark that Theorem 11 holds over *any* field since all of the arguments hold over any characteristic.

4. Families of graphs with bounded well-covered dimension.

In this section, we shall determine (in polynomial time) the well-covered dimension of circulant graphs of prime order and partitionable graphs; the techniques here are based in linear algebra. We begin with circulants of prime order.

We shall need some notation for maximal independent sets of a given cardinality. For a graph G, let $\mathcal{I}_t = \{I : I \text{ is a maximal independent set of } G, |I| = t\}$. Here is an upper bound that will be quite useful in this section.

LEMMA 17. *Let G be a graph G of order n, and let $t \leq \beta(G)$. Moreover, if $\mathrm{char}(\mathbf{F}) \neq 0$, suppose that $\gcd(t, \mathrm{char}(\mathbf{F})) = 1$. Let d_t be the dimension of the subspace of $\mathbf{F}^n$ generated by the characteristic vectors of $\mathcal{I}_t$. Then $\mathrm{wcdim}(G, \mathbf{F}) \leq n - d_t + 1$. Moreover, if $d_t = n$, then the only possible well-covered weightings are constant functions.*

Proof. If $\mathbf{w}$ is a well-covered weighting of G with sum k, then $(\mathbf{w} - \frac{k}{t}\mathbf{j})\dot{i}_t = k - k = 0$, and so $w \in \mathrm{span}(\langle \mathcal{I}_t \rangle^{\perp} \cup \{\mathbf{j}\})$. It follows that $\mathrm{wcdim}(G, \mathbf{F})$ is at most the dimension of $\mathrm{span}(\langle \mathcal{I}_t \rangle^{\perp} \cup \{\mathbf{j}\})$. The dimension of the latter is at most $(n - d_t) + 1$, and so it follows that $\mathrm{wcdim}(G, \mathbf{F}) \leq n - d_t + 1$. If $d_t = n$, then $\mathrm{span}(\langle \mathcal{I}_t \rangle^{\perp} \cup \{\mathbf{j}\}) = \mathrm{span}(\{\mathbf{j}\})$, so the only possible well-covered weightings are constant functions. $\square$

THEOREM 18. *Let G be a circulant graph with order p, a prime. If G is not totally disconnected, then* $\mathrm{wcdim}(G) = 1$ *if G is well-covered and equals* 0 *otherwise.*

Proof. Let $V(G) = \{0, 1, 2, \ldots, p-1\}$. Let S be a maximum independent set that contains 0. Since G is not totally disconnected, then $S \neq \{0, 1, 2, \ldots, p-1\}$. Note that $S_i = \{i + j \bmod p : j \in S\}$ is a maximum independent set for $i = 0, 1, \ldots, p-1$ and that $S_i \neq S_j$ for $i \neq j$. Let A be the incidence matrix where the rows are indexed by S_i and the columns by $V(G)$. A is clearly a circulant matrix. As in section 2, the determinant of A is given by

$$\prod \sum_{A(0,i)=1} x^{i-1} \qquad (**)$$

over all x that are pth roots of unity. Suppose (to reach a contradiction) that for some pth root of unity, q, $\sum_{A(0,i)=1} q^{i-1} = 0$ (we follow the argument given in [4] for vanishing sums of roots of unity). Then the automorphism $\omega \to \omega^j$ of $\mathbb{Q}[\omega]$ shows that $\sum_{A(0,i)=1} \omega^{i-1} = 0$ for *all* primitive pth roots of unity. We now sum $(**)$ over all primitive pth roots of unity, noting that, for any primitive pth root of unity and any $1 \leq j \leq p-1$, the sum of the jth power of the primitive pth roots of unity is -1 (since this is equal to the sum of the primitive pth roots of unity). Thus

$$\begin{aligned} 0 &= \sum_{j=1}^{p-1} \sum_{A(0,i)=1} \omega_j^{i-1} \\ &= \sum_{A(0,i)=1} \sum_{j=1}^{p-1} \omega_j^{i-1} \\ &= (p-1) + (|\{i : A(0,i) = 1\}| - 1)(-1). \end{aligned}$$

Therefore, the number of nonzero terms in the first row of A must be p, implying that G is totally disconnected, which is a contradiction. Since $\det(A) \neq 0$, then A is invertible and so the row space of A has dimension p. From Lemma 17, it follows that $\mathrm{wcdim}(G) \leq p - p + 1 = 1$. If G is well-covered, then $\mathbf{j}$ is a well-covered weighting. If G is not well-covered, then the only well-covered weighting is the all-zero weighting. □

5. Conclusion.

The results in the previous sections give rise to a number of questions.

PROBLEM 19. *Is it possible to give a structural characterization of anti-well-covered graphs of positive dimension? Indeed, is there a polynomial algorithm to recognize such anti-well-covered graphs?*

PROBLEM 20. *As indicated in* [2]*, the same questions can be asked of hypergraphs. Can the well-covered dimension of matroids be calculated in polynomial time?*

We can show that the well-covered dimension of a graphic matroid of a graph G is equal to the number of blocks of G.

REFERENCES

[1] Y. CARO, M. N. ELLINGHAM, AND J. E. RAMEY, *Local structure when all maximal independent sets have equal weight*, SIAM J. Discrete Math., 11 (1998), pp. 644–654.

[2] Y. CARO AND R. YUSTER, *The uniformity space of hypergraphs and its applications*, Discrete Math., 202 (1999), pp. 1–19.

[3] G. CHARTRAND AND L. LESNIAK, *Graphs and Digraphs*, Wadsworth and Brooks/Cole, Monterey, CA, 1986.

[4] J. H. Conway and A. J. Jones, *Trigonometric diophantine equations (on vanishing sums of roots of unity)*, Acta Arith., 30 (1976), pp. 229–240.

[5] D. G. Corneil, Y. Perl, and L. K. Stewart, *A linear recognition algorithm for cographs*, SIAM J. Comput., 14 (1985), pp. 926–934.

[6] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[7] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, Academic Press, Orlando, FL, 1985.

[8] M. D. Plummer, *Some covering concepts in graphs*, J. Combin. Theory, 8 (1970), pp. 91–98.

SIAM J. Discrete Math.
Vol. 19, No. 4, pp. 966–984

ON THE COMPLEXITY OF SOME ENUMERATION PROBLEMS FOR MATROIDS*

L. KHACHIYAN[†], E. BOROS[‡], K. ELBASSIONI[§], V. GURVICH[‡], AND K. MAKINO[¶]

Abstract. Let M be a matroid defined by an independence oracle on ground set S, and let $A \subseteq S$. We present an incremental polynomial-time algorithm for enumerating all minimal (maximal) subsets of S which span (do not span) A. Special cases of these problems include the generation of bases, circuits, hyperplanes, flats of given rank, circuits through a given element, generalized Steiner trees, and multiway cuts in graphs, as well as some other applications. We also consider some tractable and NP-hard generation problems related to systems of polymatroid inequalities and (generalized) packing and spanning in matroids.

Key words. matroid, base, circuit, flat, hyperplane, enumeration, hypergraph, Steiner tree, multiway cut, incremental polynomial time

AMS subject classifications. 05B35, 05A99, 68Q25

DOI. 10.1137/S0895480103428338

1. Introduction.

We assume familiarity with standard terminology of matroid theory; see, e.g., [23] for a thorough introduction. Let M be a matroid on ground set S of cardinality $|S| = n$. Throughout the paper we consider M to be defined by an *independence oracle*, i.e., an algorithm $\mathcal{I}$ which, given a subset X of S, can determine in unit time whether or not X is independent in M. This implies that the rank of any set $X \subseteq S$,

$$r(X) = \max\{|I| : I \text{ independent subset of } X\},$$

and, in particular, the rank of the matroid $r(M) \overset{\text{def}}{=} r(S)$ can be determined in $O(n)$ time by the well-known greedy algorithm. Hence, the rank of X in the dual matroid M^*

$$r^*(X) = r(S \setminus X) + |X| - r(M) \tag{1}$$

can also be computed in $O(n)$ time. In particular, $\mathcal{I}$ can be used as an independence oracle for the dual matroid.

For a subset X of S, let $Span(X) = \{y \in S \mid r(X \cup y) = r(X)\}$. A set X is said to span another set Y if $Y \subseteq Span(X)$. In this paper we consider the following enumeration problem.

*Received by the editors May 23, 2003; accepted for publication (in revised form) June 28, 2005; published electronically January 6, 2006. This research was supported in part by National Science Foundation grant IIS-0118635 and by DIMACS, the National Science Foundation's Center for Discrete Mathematics and Theoretical Computer Science.

http://www.siam.org/journals/sidma/19-4/42833.html

[†]While this paper was reviewed, our friend and co-author Leonid Khachiyan passed away with tragic suddenness. Department of Computer Science, Rutgers University, 110 Frelinghuysen Road, Piscataway, NJ 08854-8003.

[‡]RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854-8003 (boros@rutcor.rutgers.edu, gurvich@rutcor.rutgers.edu).

[§]Max-Planck-Institut für Informatik, Saarbrücken, Germany (elbassio@mpi-sb.mpg.de, elbassio@rutcor.rutgers.edu).

[¶]Division of Mathematical Science for Social Systems, Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka, 560-8531, Japan (makino@sys.es.osaka-u.ac.jp).

(P1) *Given a matroid M, defined by an independence oracle on ground set S, and two nonempty (and not necessarily disjoint) subsets D and A of S, enumerate all minimal subsets $X \subseteq D$ which span A.*

We denote the family of all such minimal spanning sets by $\mathcal{SPAN}(D, A)$, and we show that $\mathcal{SPAN}(D, A)$ can be generated in incremental polynomial time.

Theorem 1. *Computing k elements of $\mathcal{SPAN}(D, A)$ can be carried out in $poly(n, k)$ time for each $k \leq |\mathcal{SPAN}(D, A)|$.*

As will be discussed in section 2.4, problem (P1) is equivalent to another enumeration problem.

(P2) *Given a matroid M on S and two nonempty subsets D and A of S, enumerate all maximal subsets $X \subseteq D$ such that $A \not\subseteq Span(X)$.*

Corollary 1. *The enumeration problem for* (P2) *can also be solved in incremental polynomial time.*

Note that in the statements of problems (P1) and (P2), Theorem 1, and Corollary 1 we can assume without loss of generality that $S = D \cup A$.

Unlike problems (P1) and (P2), we show that the following two enumeration problems are intractable: *Given a matroid M on S and two nonempty disjoint subsets $D, A \subset S$, enumerate*

(P3) *all minimal subsets X of D such that $Span(X) \cap A \neq \emptyset$.*

(P4) *all maximal subsets $X \subseteq D$ such that $Span(X) \cap A = \emptyset$.*

Theorem 2. *Let D and A be given sets of vectors in a vectorial matroid M.*

(i) *Given a collection $\mathcal{X}$ of minimal subsets $X \subseteq D$ for which $Span(X) \cap A \neq \emptyset$, it is NP-complete to decide whether $\mathcal{X}$ can be extended.*

(ii) *Given a collection $\mathcal{X}$ of maximal subsets X of D satisfying $Span(X) \cap A = \emptyset$, it is NP-complete to decide whether $\mathcal{X}$ can be extended.*

As we will see in section 3, part (i) of Theorem 2 holds for vectorial matroids over an arbitrary field $\mathbf{F}$. Our proof of part (ii) requires that $\mathbf{F}$ be large (of characteristic 0 or linear in $n = |S|$). In particular, we do not know whether part (ii) of Theorem 2 holds for binary matroids. Note, however, that for binary matroids problem (P4) includes as a special case the well-known *hypergraph dualization problem* [8, 9, 10]: enumerate all maximal independent vertex sets for a given hypergraph $H \subseteq 2^S$. (To see this, let D and A be the sets of characteristic vectors of all vertices and all hyperedges of H, respectively.) No incremental polynomial-time algorithm for dualization of arbitrary hypergraphs is currently known. Clearly, Theorem 2 implies that for vectorial matroids, problems (P3) and (P4) cannot be solved in incremental (or output) polynomial time unless $P = \mathrm{NP}$.

It is also worth mentioning that problems (P1) and (P2) are mutually dual in the sense that they call for enumerating minimal/maximal subsets X of S that satisfy/violate a certain monotone set property (specifically, X spans A). Problems (P3) and (P4) are also mutually dual (with the common monotone property $Span(X) \cap A \neq \emptyset$). In general, the complexities of two mutually dual enumeration problems may differ substantially. For instance, one problem may be NP-hard while the other is solvable in incremental polynomial time (see, e.g., [11] and also the examples in sections 4 and 5). In particular, the duality of problems (P3) and (P4) by no means implies that parts (i) and (ii) of Theorem 2 are equivalent. This is why we prove parts (i) and (ii) of Theorem 2 separately.

Before proceeding further, we consider some special cases of Theorem 1 and Corollary 1. Note that since $A \subseteq Span(X)$ implies $Span(A) \subseteq Span(X)$, we can assume, without loss of generality, that in problems (P1) and (P2) the set A is *a flat*, i.e.,

$A = Span(A)$. Note also that the minimal sets X enumerated in problem (P1) are independent, i.e., $r(X) = |X|$, while the maximal sets in problem (P2) are flats.

1.1. Bases. When $A = D = S$, then $\mathcal{SPAN}(S, S)$ is the set of all *bases* of M, i.e., the collection $\mathcal{B}(M)$ of all minimal subsets $B \subseteq S$ that span S. It is a folkloric result that, for this special case, all elements of $\mathcal{SPAN}(S, S) = \mathcal{B}(M)$ can be enumerated with *polynomial delay*, i.e., in *poly*(n) time per each generated base. This can be done, for instance, by traversing the connected "supergraph" $\mathcal{G} = (\mathcal{B}(M), \mathcal{E})$ in which two "vertices" $B, B' \in \mathcal{B}(M)$ are connected by an edge in $\mathcal{E}$ if and only if B and B' can be obtained from each other by exchanging a pair of elements, i.e., when $|B \setminus B'| = |B' \setminus B| = 1$. The connectivity of $\mathcal{G}$ then follows from the well-known *base axiom: If* $B, B' \in \mathcal{B}(M)$ *and* $x \in B' \setminus B$, *then* $(B \cup y) \setminus x \in \mathcal{B}(M)$ *for some* $y \in B \setminus B'$.

1.2. Hyperplanes. Assuming as before $A = D = S$, problem (P2) calls for enumerating all *hyperplanes* of M, i.e., all flats of rank $r(M) - 1$. Seymour showed that all hyperplanes of a matroid can be enumerated in incremental polynomial time [21]. This also implies an incremental polynomial time algorithm for enumerating all flats of a given rank t because such flats are the hyperplanes of the truncated matroid M_{t+1} whose rank function is defined by $r_{t+1}(X) = \min\{r(X), t + 1\}$.

1.3. Circuits. The bases of the dual matroid M^* are the complements to the bases of M: $\mathcal{B}(M^*) = \{S \setminus B \ : \ B \in \mathcal{B}(M)\}$. Let $\mathcal{C}(M)$ be the set of all *circuits* of M, i.e., the collection of all minimal dependent sets in M. Since each circuit of M is the complement of a hyperplane of M^* and vice versa, the enumeration of all circuits of M is equivalent to the hyperplane enumeration for the dual matroid M^*. Hence by 1.2 the set $\mathcal{C}(M)$ of all circuits of M can be enumerated in incremental polynomial time. When M is the cycle matroid of a given graph $G = (V, E)$ and consequently $\mathcal{C}(M)$ is the family of all simple cycles of G, all elements of $\mathcal{C}(M)$ can be enumerated with polynomial delay (see, e.g., [19]). This is also true for M^*, the cocycle matroid of G, where each element of $\mathcal{C}(M^*)$ is a minimal set of edges whose removal increases the number of connected components of G (see, e.g., [18]). In general, however, no polynomial-delay algorithm is known for enumerating all circuits (equivalently, hyperplanes) of an arbitrary matroid.

1.4. Circuits through a given point. When $A = \{a\}$ consists of a single element of rank 1, and $D = S \setminus A$, then $I \in \mathcal{SPAN}(D, A)$ if and only if $I \cup \{a\}$ is a circuit containing a. Thus Theorem 1 implies that all circuits through a given element can be generated in incremental polynomial time. When M is the cycle or cocycle matroid of a connected graph $G = (V, E)$ and $a = (uv) \in E$ is an edge with endpoints $u, v \in V$, enumerating all circuits through a calls for generating all simple uv-paths or all minimal uv-cuts in G, which can again be done with polynomial delay [18]. However, for general matroids, no polynomial delay algorithm is known. Furthermore, we are not aware of an incremental polynomial time algorithm for enumerating all circuits containing $t = 2$ elements of a given matroid M. In section 2.3 we argue that this problem can be solved with polynomial delay for each fixed t when M is the cycle matroid of a graph but becomes NP-hard when t is part of the input.

1.5. Vertex enumeration. An open question in linear programming is whether there exists an efficient way to enumerate all vertices of a given polytope

$$P = \left\{ x = (x_1, \ldots, x_n) \in \Re^n \mid \sum_{i=1}^{n} a_i x_i = a, \quad x_1, \ldots, x_n \geq 0 \right\},$$

where $a, a_1, \dots, a_n$ are given d-dimensional vectors. Each vertex of P can be identified with a minimal supporting set I of coordinates $\{1, \dots, n\}$ for which the system of linear equations

$$\sum_{i \in I} a_i x_i = a \tag{2}$$

has a nonnegative, and hence positive, real solution. Dropping the nonnegativity conditions, we arrive at the problem of enumerating all minimal sets $X \subseteq \{1, \dots, n\}$ for which (2) has a real solution. This is equivalent to the enumeration of all circuits through a in the vectorial matroid $M = \{a, a_1, \dots, a_n\} \subseteq \Re^d$. By the same token, given an infeasible system of linear equations, all minimal infeasible (maximal feasible) subsystems of the given system can be enumerated in incremental polynomial time.

We now consider some other special cases of problems (P1) and (P2) when M is the cycle matroid of a (multi-)graph.

1.6. Generalized Steiner trees and point-to-point connections. Let $G = (V, E)$ be a graph with k given disjoint vertex sets $V_1, \dots, V_k \subseteq V$. A *generalized Steiner tree* is a minimal set of edges $X \subseteq E$ connecting all vertices within each set V_i; i.e., for each $i = 1, \dots, k$, all vertices of V_i must belong to a single connected component of (V, X). In particular, for $k = 1$ we obtain the usual definition of Steiner trees. When each set V_i consists of two vertices $\{u_i, v_i\}$, generalized Steiner trees are called *point-to-point* connections. For $i = 1, \dots, k$, let A_i be the collection of $\binom{|V_i|}{2}$ "new" edges that connect every pair of vertices within V_i, let $A = A_1 \cup \cdots \cup A_k$, and let M be the cycle matroid of the multigraph $(V, E \cup A)$. Then $\mathcal{SPAN}(E, A)$ is the family of all generalized Steiner trees for $V_1, \dots, V_k$ because an edge set X spans an edge e in M if and only if X contains a path connecting the endpoints of e. Thus the enumeration of generalized Steiner trees is a special case of problem (P1), and Theorem 1 implies that there is an incremental polynomial time algorithm for enumerating generalized Steiner trees. Note that in this problem, we can also replace each A_i by a spanning tree on V_i.

1.7. Multiway cuts. Let $V' \subseteq V$ be a vertex set of a graph $G = (V, E)$. A *multiway cut* is a minimal collection of edges whose removal disconnects every pair of vertices in V'; see, e.g., [22]. Note that since any edge between two vertices of V' must be included in each multiway cut, we can assume without loss of generality that V' is a stable vertex set in G. Let A be a set of $|V'| - 1$ "new" edges forming a spanning tree on V', and let M^* be the cographical matroid of $G' = (V, E')$, where $E' = E \cup A$. An edge set $X \subseteq E$ spans A in M^* if and only if $r^*(X \cup A) = r^*(X)$, where $r^*(\cdot)$ is the rank function of M^*. From (1), it follows that $r(Y) + |A| = r(Y \cup A)$, where $Y = E \setminus X$ is the complement of X in $G = (V, E)$. So if we remove X from G and start adding the edges of A to the resulting graph (V, Y), then each new edge from A should be decreasing the number of connected components. This is the same as saying that the vertices of V' are all in distinct connected components of (V, Y), i.e., that X is a multiway cut. So the enumeration problem for multiway cuts is equivalent to the enumeration of all minimal subsets X of E such that X spans A in M^*, which, by Theorem 1, can be done in incremental polynomial time.

1.8. Disjunctions of paths. This enumeration problem is dual to the enumeration of multiway cuts: Given a vertex set $V' \subseteq V$ in a graph $G = (V, E)$, enumerate all minimal subsets $X \subseteq E$ which connect some pair of vertices in V'. By Corollary

1, all such minimal paths connecting a pair of vertices in V' can be enumerated in incremental polynomial time.

The remainder of this paper is organized as follows. We prove Theorems 1 and 2 in sections 2 and 3, respectively. In section 4 we discuss some circuit and hyperplane enumeration problems for two matroids on S, and we also discuss *generalized* circuits and hyperplanes obtained by replacing some singletons of S by subsets, i.e., by performing the parallel extension of the rank function $r(X)$ for some sets $A_1, \dots, A_n \subseteq S$. We show that the enumeration problems corresponding to these variants and generalizations of circuits and hyperplanes are all NP-hard already for graphic and cographic matroids.

Finally, in section 5 we discuss similar generalizations of matroid bases and show that they can be expressed in a natural way as minimal solutions to some systems of polymatroid inequalities. Due to results obtained in [2, 4], this implies that the corresponding enumeration problems for generalized bases, including spanning, packing, and the maximal independent set problems for several matroids, can all be solved in incremental *quasi-polynomial* time $2^{\mathrm{polylog}(n,k)}$, where k is the number of generated objects.

2. Minimal spanning sets for a flat.

In this section we prove Theorem 1. For completeness, we start with an incremental polynomial-time algorithm for generating all circuits of a matroid, which is dual to the hyperplane generation algorithm suggested by Seymour [21].

2.1. Enumerating all circuits of a matroid.

Let M be a matroid defined by an independence oracle on ground set S of size n, and let $\mathcal{C}(M) \subseteq 2^S$ be the family of all circuits of M.

PROPOSITION 1 (see [21]). *$\mathcal{C}(M)$ can be enumerated in incremental polynomial time.*

Proof. Start by computing a base B^o of M. Next, for each $x \in S \setminus B^o$ there exists a unique circuit $C = C(B^o, x)$ such that $x \in C \subseteq B^o \cup x$. This circuit $C(B^o, x)$, called the *fundamental circuit* of x in the base B^o, can be computed by querying the independence oracle on at most $|B^o|$ subsets of $B^o \cup x$. Denote by $\mathcal{F}(B^o) = \{C(B^o, x) \mid x \in S \setminus B^o\}$ the set of $n - r(M)$ fundamental circuits for B^o.

The family $\mathcal{C}(M)$ of circuits of any matroid satisfies the *circuit axiom*: *If C_1 and C_2 are distinct circuits of M and $e \in C_1 \cap C_2$, then there exists a circuit C_3 such that $C_3 \subseteq (C_1 \cup C_2) \setminus e$.*

To enumerate all circuits in M, start with $\mathcal{C}' = \mathcal{F}(B^o)$ and repeatedly check whether $\mathcal{C}'$ is closed with respect to the circuit axiom. Since each violation of the circuit axiom produces a new circuit, it remains to argue that, if some system $\mathcal{C}'$ of circuits is closed with respect to the circuit axiom and $\mathcal{F}(B^o) \subseteq \mathcal{C}'$, then $\mathcal{C}' = \mathcal{C}(M)$. This follows from the fact that any set system $\mathcal{C}' \subseteq 2^S$ satisfying the circuit axiom and the Sperner condition $C_1, C_2 \in \mathcal{C} \Longrightarrow C_1 \not\subseteq C_2$ defines a matroid M' on S; see [16, 23]. By definition, the bases of M' are all maximal independent sets for $\mathcal{C}'$, i.e., all those maximal subsets of S which contain no set in $\mathcal{C}'$. In our case $\mathcal{C}' \subseteq \mathcal{C}(M)$, and hence $\mathcal{C}'$ is Sperner by definition. Furthermore, since $\mathcal{C}'$ contains the fundamental system of circuits for $B^o \in \mathcal{B}(M)$, it follows that B^o is also a base of M', implying that the ranks of M and M' are equal. Let $C \in \mathcal{C}(M)$ be an arbitrary circuit of M; then C is the fundamental circuit for some base $B \in \mathcal{B}(M)$ and some element $x \in S \setminus B$, i.e., $C = C(B, x)$. Since B is independent in M' and $|B| = r(M) = r(M')$, it follows that $B \in \mathcal{B}(M')$. Now M' must also contain a unique fundamental circuit $C' = C'(B, x)$. Since any circuit of M' is also a circuit of M, it must be the case that $C = C(B,x) = C'(B,x)$, which shows that $C \in \mathcal{C}' = \mathcal{C}(M')$. □

2.2. Circuits through a given element. We next prove the special case of Theorem 1 for $|A| = 1$.

Proposition 2. *Let M be a matroid with ground set S, let $a \in S$, and let $\mathcal{C}(M,a)$ be the set of circuits C of M such that $a \in C$. Assuming that M is defined by an independence oracle, all elements of $\mathcal{C}(M,a)$ can be enumerated in incremental polynomial time.*

Proof. Two elements $x, y \in S$ are said to be *connected* in M if either $x = y$ or there is a circuit $C \in \mathcal{C}(M)$ containing both x and y. It is well known that this definition results in an equivalence relation on S, each equivalence class of which is called a connected component of M. In particular, M is connected if S is the only connected component of M. It is also known that, given an independence oracle for M, the connected components of M can be determined in polynomial time [1].

Returning to the problem of enumerating all circuits of M through the given element a, we observe that all such circuits must belong to the connected component of M which contains a. So we may replace S by this connected component and assume without loss of generality that M is connected.

Given a set $X \subseteq S$, let

$$D(X) = X \setminus \bigcap\{C \in \mathcal{C}(M,a) \mid C \subseteq X\},$$

where as before $\mathcal{C}(M,a)$ denotes the set of all circuits containing a. Lehman's theorem [16, 23] asserts that for any connected matroid M the circuits of M not containing a are precisely the minimal sets of the form $D(C_1 \cup C_2)$, where C_1 and C_2 are distinct members of $\mathcal{C}(M,a)$. Hence for any connected matroid M we have the following bound:

$$|\mathcal{C}(M)| \le |\mathcal{C}(M,a)|\big(|\mathcal{C}(M,a)| + 1\big)/2.$$

This bound and Proposition 1 readily imply that all circuits in $\mathcal{C}(M,a)$ can be enumerated in output polynomial time *poly*$(n, |\mathcal{C}(M,a)|)$ by simply generating all circuits in $\mathcal{C}(M)$ and discarding those that do not pass through a. In fact, since our enumeration problem is self-reducible, the above bound also implies an incremental polynomial-time algorithm. To see this, assume that we wish to enumerate a given number k of circuits in $\mathcal{C}(M,a)$ or to list all of them if $k \ge |\mathcal{C}(M,a)|$. Since for each integer $k' \le |\mathcal{C}(M)|$ we can obtain k' circuits in $\mathcal{C}(M)$ in *poly*(n,k') time, we can decide whether or not $k \ge |\mathcal{C}(M,a)|$ by attempting to generate $k' = k(k+1)/2$ circuits in $\mathcal{C}(M)$, in time bounded by a polynomial in n and k. If we discover that $|\mathcal{C}(M)| \le k(k+1)/2$ by producing all circuits in $\mathcal{C}(M)$, then we also have the entire set $\mathcal{C}(M,a)$. Suppose now that we have computed $k(k+1)/2$ circuits in $\mathcal{C}(M)$ but fewer than k of them pass through a. Let $b \ne a$ be another element of S. Delete b and compute the connected component S' which contains a in the matroid M restricted to $S \setminus b$. Note that any circuit of $\mathcal{C}(M,a)$ which does not contain b must belong to S'. So we may apply the same procedure to the connected matroid M' obtained by restricting M to S' and either obtain all circuits of $\mathcal{C}(M,a)$ which avoid b, or conclude that the number of such circuits exceeds k. Since in the latter case we can reduce the size of S by removing b for good (as long as we are not required to produce more than k circuits of $\mathcal{C}(M,a)$), we may now assume without loss of generality that for each element $b \ne a$ we have obtained all the circuits in $\mathcal{C}(M,a)$ which avoid b. This means that in time polynomial in n and k we can produce all circuits in $\mathcal{C}(M,a)$ which skip some element of S. Unless S itself is the only element of $\mathcal{C}(M,a)$, this gives the entire set $\mathcal{C}(M,a)$. ▯

2.3. Circuits through t elements. It is natural to ask what is the complexity of enumerating all circuits of M which contain a given set $A = \{a_1, \dots, a_t\}$ of $t \geq 2$ elements of S. We digress from the proof of Theorem 1 and argue that this problem is NP-hard when t is part of the input but can be solved with polynomial delay if $t = |A|$ is fixed and M is the cycle matroid of a given graph $G = (V, E)$. As mentioned in the introduction, we are not aware of an efficient algorithm for listing all circuits through $t = \text{const} \geq 2$ elements of an arbitrary matroid.

Let M be the cycle matroid of G so that the circuits of M are the simple cycles of G. An edge set A may be contained in a simple cycle only if A itself is a simple cycle or A is a union of k pairwise vertex disjoint simple paths $P_1, \dots, P_k$ for some positive integer $k \leq t$. All simple cycles containing $P_1, \dots, P_k$ can be enumerated with polynomial delay via lexicographic backtracking [19] by growing and merging these partial paths (so that their number continually decreases). Hence backtracking listing algorithms reduces the enumeration of simple cycles containing $a_1, \dots, a_t$ to the following decision problem: *Does there exist a simple cycle in G which contains k given disjoint paths $P_1, \dots, P_k$?*

When k is fixed, by considering all possible permutations and reversals of $P_1, \dots, P_k$ the latter problem can in turn be polynomially reduced to the well-known *disjoint-path problem: Given k pairs of vertices $\{u_i, v_i\}$, $i = 1, \dots, k$, of a graph, can these pairs be connected by k pairwise vertex disjoint paths?*

Even though the disjoint-path problem is NP-complete, when k is part of the input (see [13]), it is known [20] to be solvable in polynomial time for each fixed k. Hence all simple cycles through $t = \text{const}$ edges can be enumerated with delay bounded by a polynomial in the size of the input graph.

However, if $t = |A|$ is part of the input, then the problem of enumerating all simple cycles through t edges of a graph becomes NP-complete. In fact, given a graph $G = (V, E)$ and a (large) matching $A \subset E$, it is NP-hard to decide whether G has *any* simple cycle containing A. This can be seen from the following argument. Given a graph $H = (U, E)$, substitute an edge e_u for each vertex $u \in U$. Then, unless G consists of a single edge, the resulting graph $G = P_2 \times H$ has a simple cycle through the matching $A = \{e_u \ : \ u \in U\}$ if and only if the original graph H is Hamiltonian, a condition which is NP-complete to verify.

Finally, similar to minimal cycles, it can be shown that the enumeration of all minimal cuts through t edges is NP-hard when t is part of the input. These results also indicate that it is NP-complete to decide whether a cycle or cocycle matroid M has a hyperplane avoiding all elements of a given set A.

2.4. Proof of Theorem 1. We now complete the proof of Theorem 1. Let, as before, M be a matroid on S, and let D and A be two nonempty subsets of S. We use a new element α to represent A. Specifically, let M_α be the matroid on $D \cup \alpha$ with the following rank function:

$$\rho(X) = \begin{cases} r(X) & \text{if} \quad \alpha \notin X, \\ \max\{r((X \setminus \alpha) \cup a) \mid a \in A\} & \text{otherwise}. \end{cases} \tag{3}$$

It is easy to check that M_α is indeed a matroid. When M is a vectorial matroid over a large field, α can be interpreted as the "general linear combination" of all elements of A; in general, $\rho(X)$ is the so-called *principal extension of* $r(X)$ *on* A *with value* 1 (see, e.g., [17]).

When $I \in \mathcal{SPAN}(D, A)$, then $I \cup \alpha$ is a circuit in M_α, and conversely, for any circuit C in M_α containing α, the set $C \setminus \alpha$ belongs to $\mathcal{SPAN}(D, A)$. Hence the enumeration problem for $\mathcal{SPAN}(D, A)$ is equivalent with that for the set of all circuits through α in M_α. Given an independence oracle for M, the rank function (3) of the extended matroid can be trivially evaluated in oracle-polynomial time. Therefore Theorem 1 directly follows from Proposition 2. $\square$

We mention in passing that since $\mathcal{SPAN}(S, S)$ is the set of bases of M, the proof of Theorem 1 shows that the enumeration of all bases of a matroid M can be reduced to the enumeration of all circuits (equivalently, all circuits through a given element) of another matroid.

Proof of Corollary 1. Given two nonempty subsets $D, A \subseteq S$, we wish to enumerate all maximal subsets $X \subseteq D$ such that $A \not\subseteq \mathit{Span}(X)$. Consider the matroid M_α constructed in the proof of Theorem 1. The ground set of M_α is $D \cup \{\alpha\}$, and a set $X \subseteq D$ does not span α if and only if $A \not\subseteq \mathit{Span}(X)$; see (3). On the other hand $X \subseteq D$ does not span α in M_α if and only if $Y = D \setminus X$ spans α in the dual matroid M^*_α. By Theorem 1, we can enumerate all minimal Y spanning α in M^*_α in incremental polynomial time. $\square$

3. Proof of Theorem 2. *Part* (i). We reduce our enumeration problem from the CNF satisfiability problem: Given m clauses (i.e., disjunctions) $C_1, \ldots, C_m$ of some literals drawn from $L = \{x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n\}$, does there exist a truth assignment of $x_1, \ldots, x_n$ that satisfies the conjunctive normal form $\phi(x_1, \ldots, x_n) = C_1 \wedge \cdots \wedge C_m$?

Let $|C_j|$ denote the number of literals in clause C_j. We construct a graph $G = (V, E)$ on $|V| = 2mn - n + m + 1 + \sum_{j=1}^m |C_j|$ vertices and $|E| = 2mn + 2\sum_{j=1}^m |C_j|$ edges as follows. For each positive literal x_i appearing in clause C_j, we introduce three edges $Y_{ij}, Y'_{ij}, Y''_{ij} \in E$, and for each negative literal $\bar{x}_i$ appearing in clause C_j, we introduce three edges $Z_{ij}, Z'_{ij}, Z''_{ij} \in E$. These edges are connected in G as follows:

$$G = v_0 \ P_1 \ v_1 \ P_2 \ v_2 \ldots \ v_{n-1} \ P_n \ v_n \ P'_1 \ v'_1 \ P'_2 \ v'_2 \ldots \ v'_{m-1} \ P'_m \ v'_m,$$

where $v_0, v_1, \ldots, v_n = v'_0, v'_1, \ldots, v'_{m-1}, v'_m$ are $n + m + 1$ distinct vertices; each P_i consists of two parallel chains, $Y_{i1}, \ldots, Y_{im}$ and $Z_{i1}, \ldots, Z_{im}$ of m edges in each, connecting v_{i-1} and v_i; and each P'_j consists of $|C_j|$ parallel chains of two edges labeled by either the variables Y'_{ij}, Y''_{ij} or by Z'_{ij}, Z''_{ij}, depending on whether literal x_i or $\bar{x}_i$ appears in C_j. See Figure 1 for a small example.

For each edge $e = (uv) \in E$, let us now select an orientation, say $u \to v$, and let $\chi(e) \in \{0, \pm 1\}^V$ be the vector with the components $\chi_u = -1$, $\chi_v = +1$, and $\chi_w = 0$ for $w \notin \{u, v\}$, where the scalars $0, -1$, and $+1$ are drawn from some field $\mathbf{F}$ (we do not exclude the binary case $+1 = -1$). We will also denote by $\mu = \chi(v_0, v'_m)$ the vector whose components are given by $\mu_{v_0} = -1$, $\mu_{v'_m} = +1$, and $\mu_w = 0$ otherwise. Let $\mathcal{B}$ be the following set of $\sum_{j=1}^m |C_j|$ pairs of edges:

$$\mathcal{B} = \{\{Y_{ij}, Z'_{ij}\} \ : \ \text{literal } x_i \in C_j\} \ \cup \ \{\{Z_{ij}, Y'_{ij}\} \ : \ \text{literal } \bar{x}_i \in C_j\},$$

and let

$$D = \{\chi(e), \ e \in E\} \ \text{ and } \ A = \{\mu\} \cup \{\chi(e) + \chi(e') \ : \ \{e, e'\} \in \mathcal{B}\}.$$

Finally, let M be the vectorial matroid over $\mathbf{F}$ on the ground set $S = D \cup A$. Denote by $\mathcal{F}(D, A)$ the family of all minimal subsets of D that span at least one vector

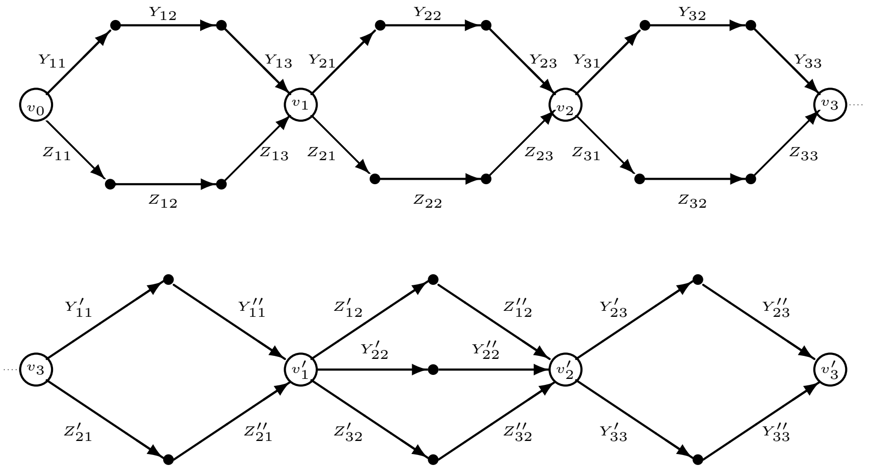


FIG. 1. *The directed graph used in the proof of* (i) *of Theorem* 2 *corresponding to the CNF* $\phi(x_1, x_2, x_3) = (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge (x_2 \vee x_3)$.

from $\mathcal{A}$. It is easy to see that each vector $\chi(e) + \chi(e'), \{e, e'\} \in \mathcal{B}$ in $A \setminus \{\mu\}$ is spanned by exactly $|C_j| - 1$ sets from $\mathcal{F}(D, A)$, where C_j is the clause containing e'. All such sets in $\mathcal{F}(D, A)$ can be easily listed, and their number is bounded by $O(\sum_{j=1}^{m} |C_j|^2) \leq poly(n, m)$. Deciding whether $\mathcal{F}(D, A)$ contains other sets is NP-complete because any additional set X in $\mathcal{F}(D, A)$ must span μ and correspond to a v_o-v'_m path that contains no pair of mutually negating literals. This means that X can be transformed into a satisfying assignment for the input CNF ϕ. □

Part (ii). Again, we use a reduction from the CNF satisfiability problem, specifically, from the well-known NP-complete 3-SAT problem. As before, we denote by $L = \{x_1, x_2, \dots, x_n, \bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\}$ the set of literals, and let $\phi(x_1, \dots, x_n) = C_1 \wedge \cdots \wedge C_m$ be a given cubic CNF, i.e., $|C_i| = 3$ for $i = 1, \dots, m$. It will be convenient to introduce a linear ordering $\prec$ on L, say $x_1 \prec \bar{x}_1 \prec x_2 \prec \bar{x}_2 \prec \cdots \prec x_n \prec \bar{x}_n$, and assume that $C_i = \{\ell_{i1}, \ell_{i2}, \ell_{i3}\}$, where $\ell_{i1} \prec \ell_{i2} \prec \ell_{i3}$ for $i = 1, 2, \dots m$.

Let $V = \{0, 1, \dots, n\}$, and let $e^j \in \{0, 1\}^V$ denote the jth unit vector, i.e., $e^j_k = 1$ if and only if $j = k$, for $j, k \in V$. We start by associating to every literal $\ell \in L$ a $|V|$-dimensional $\{0, 1\}$-vector d^{ℓ} by defining, for each $j = 1, \dots, n$,

$$d^{x_j} = e^j \quad \text{ and } \quad d^{\bar{x}_j} = e^0 + e^j.$$

Next, we let $w_1 = 2(n+1)$, $w_2 = 4(n+1)$, and $w_3 = 8(n+1)$, and we associate to each clause $C_i = \{\ell_{i1}, \ell_{i2}, \ell_{i3}\}$ a set of $n+1$ integer vectors $a^i(\lambda)$ of the following form:

$$a^i(\lambda) = \sum_{k=1}^{3} w_k d^{\bar{\ell}_{ik}} + \sum_{j=1}^{n} e^j + \lambda e^0,$$

where $\lambda = 0, \dots, n$. Finally, let us define the sets

$$D = \{d^{\ell} \mid \ell \in L\} \quad \text{ and } \quad A = \{a^i(\lambda) \mid i = 1, \dots, m,\ \lambda = 0, 1, \dots, n\}$$

and consider the vectorial matroid M on $S = D \cup A$ over any field $\mathbf{F}$ in which $0, 1, \dots, 15(n+1)$ are distinct constants. Denote by $\mathcal{F}(D, A)$ the family of all maximal subsets of D that span no vector in A.

Our first observation is that whenever a set $Y \subseteq D$ contains neither d^{x_j} nor $d^{\bar{x}_j}$ for some $j \in \{1, \dots, n\}$, then $\mathit{Span}(Y) \cap A = \emptyset$ because the jth component of any linear combination of the vectors of Y is zero, while the jth component of any vector in A is nonzero. Furthermore, if $Y \cap \{d^{x_j}, d^{\bar{x}_j}\} \neq \emptyset$ and $|Y| > n$, then Y spans the entire linear space $\mathbf{F}^V$ (and hence $\mathit{Span}(Y) \cap A \neq \emptyset$) because $\mathit{rank}(Y) = n+1$ for such a set of vectors. Thus, it follows that

$$\mathcal{X} \doteq \{D - \{d^{x_j}, d^{\bar{x}_j}\} \mid j = 1, 2, \dots, n\} \subseteq \mathcal{F}(D, A) \tag{4}$$

and that $Y \in \mathcal{F}(D, A) \setminus \mathcal{X}$ implies

$$|Y \cap \{d^{x_j}, d^{\bar{x}_j}\}| = 1 \quad \text{for} \quad j = 1, \dots, n. \tag{5}$$

Every subset $Y \subseteq D$ satisfying (5) naturally encodes a truth assignment $x(Y)$ of the n input variables, where $x_j(Y) = 1$ for all those components $j \in \{1, \dots, n\}$ for which $d^{x_j} \in Y$, and $x_j(Y) = 0$ whenever $d^{\bar{x}_j} \in Y$. Now we claim that $\mathcal{F}(D, A) \setminus \mathcal{X} \neq \emptyset$ if and only if ϕ is satisfiable. To see this we show that a truth assignment $x = x(Y)$ is a satisfying assignment for ϕ if and only if $\mathit{Span}(Y) \cap A = \emptyset$, which by (4) is equivalent to $Y \in \mathcal{F}(D, A) \setminus \mathcal{X}$. For this, suppose first that for a set $Y \subseteq D$ satisfying (5) we have

$$a^i(\lambda) \in \mathit{Span}(Y) \tag{6}$$

for some $i \in \{1, \dots, m\}$ and $\lambda \in \{0, \dots, n\}$. The n vectors in Y always form a matrix whose last n rows are the identity matrix of order n. Hence there is only one possible linear combination of the vectors in Y which could be equal to $a^i(\lambda)$. By definition, the last n components of $a^i(\lambda)$ contain $n-3$ ones plus three other components equal to w_1+1, w_2+1, and w_3+1. These are exactly the components corresponding to the three variables in the conjunction C_i. Furthermore, since the weights w_1, w_2, and w_3 are large, by looking at the 0th component of $a^i(\lambda)$ we conclude that Y must contain the three vectors $d^{\bar{\ell}_{ik}}$ corresponding to the *negations* of the literals of $C_i = \{\ell_{i1}, \ell_{i2}, \ell_{i3}\}$. But by (5) this clearly means that $x(Y)$ violates C_i. In other words, if Y encodes a satisfying truth assignment for ϕ, then Y cannot span any vector in A, and hence $Y \in \mathcal{F}(D, A) \setminus \mathcal{X}$. The converse implication also holds true: any $Y \in \mathcal{F}(D, A) \setminus \mathcal{X}$ encodes a satisfying assignment for ϕ. This is because $\phi(x(Y)) = 0$ would imply that $x(Y)$ violates some clause $C_i = \{\ell_{i1}, \ell_{i2}, \ell_{i3}\}$ and consequently that Y contains the set $D(C_i) = \{d^{\bar{\ell}_{i1}}, d^{\bar{\ell}_{i2}}, d^{\bar{\ell}_{i3}}\}$. Now it is easy to see that the linear combination

$$\sum_{k=1}^{3} (w_i + 1) d^{\bar{\ell}_{ik}} + \sum_{d \in Y \setminus D(C_i)} d \ \in \ \mathit{Span}(Y)$$

must coincide with $a^i(\lambda)$ for some $\lambda \in \{0, 1, \dots, n\}$, contradicting the selection of Y. □

We close this section by stating problems (P3) and (P4) for the dual matroid of M: *Given two disjoint sets $D, A \subset S$, enumerate*

(P3)* *all minimal sets $X \subseteq D$ such that $a \in \mathit{Span}(X \cup (A \setminus a))$ for all $a \in A$.*

(P4)* *all maximal subsets $X \subseteq D$ such that $a \notin \mathit{Span}(X \cup (A \setminus a))$ for some $a \in A$.*

Since the dual matroid for an explicitly given vectorial matroid over some field is again an explicitly given vectorial matroid over the same field, the enumeration problems (P3)* and (P4)* are also NP-hard for vectorial matroids over large fields.

4. Circuits in two matroids, and generalized circuits and hyperplanes. Let M_1 and M_2 be two matroids on S with rank functions $r_1(X)$ and $r_2(X)$. It is known that the minimum of the submodular function $r_1(X)+r_2(S\setminus X)$ for all $X \subseteq S$ gives the maximum cardinality of a set I independent in both M_1 and M_2 and that this minimum can be computed in polynomial time [7]. In particular, when the ranks of M_1 and M_2 are equal, one can determine in polynomial time whether M_1 and M_2 share a common base, i.e., $\mathcal{B}(M_1) \cap \mathcal{B}(M_2) \neq \emptyset$. In fact, using this as a subroutine for backtracking on matroids obtained by deleting and contracting elements of S, all bases in $\mathcal{B}(M_1) \cap \mathcal{B}(M_2)$ can be enumerated with polynomial delay.

In contrast to this result, deciding whether M_1 and M_2 contain a common circuit is NP-hard already when M_1 is the cycle matroid of some graph $G = (V, E)$ and M_2 is the uniform matroid on E whose bases are all subsets of size $r = |V| - 1$. In this case, $\mathcal{C}(M_1) \cap \mathcal{C}(M_2) \neq \emptyset$ if and only if G is Hamiltonian. A similar argument for the NP-complete maximum cut problem shows that testing if $\mathcal{C}(M_1) \cap \mathcal{C}(M_2) \neq \emptyset$ remains NP-hard when M_1 is the cocycle matroid of a graph $G = (V, E)$ and M_2 is again a uniform matroid on E.

Of course, given two matroids M_1 and M_2 on S, one can always enumerate all elements of $\mathcal{C}(M_1) \cup \mathcal{C}(M_2)$ in incremental polynomial time due to Theorem 1. Note, however, that deciding whether a given set $C \in \mathcal{C}(M_1) \cup \mathcal{C}(M_2)$ is *maximal* in $\mathcal{C}(M_1) \cup \mathcal{C}(M_2)$ is NP-hard, in general. This is because for any set $A \subseteq S$ we may choose M_2 to be the matroid for which A is the only circuit, and then deciding whether A is maximal becomes equivalent with determining if M_1 has a circuit containing A (see section 2.3). Perhaps more surprisingly, for two matroids M_1 and M_2 on S, enumerating all *minimal* elements of $\mathcal{C}(M_1) \cup \mathcal{C}(M_2)$ may also be hard.

PROPOSITION 3 (see [3]). *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two connected planar graphs on a common set E of n edges (i.e., both E_1 and E_2 are labeled in a one-to-one way by the elements of E). Furthermore, let M_1 and M_2 be cocycle matroids on E, corresponding to G_1 and G_2. Then it is NP-hard to enumerate $\mathcal{MIN}\{\mathcal{C}(M_1) \cup \mathcal{C}(M_2)\}$, the collection of all minimal sets $C \subseteq E$ which disconnect at least one of the graphs G_1 or G_2.*

By considering the dual planar graphs for G_1 and G_2, it can be shown that Proposition 3 also holds for the cycle matroids of G_1 and G_2. Specifically, it is NP-hard to enumerate all minimal edge sets $X \subseteq E$ which form a cycle in G_1 or G_2.

We close this section with yet another generalization of the notion of a circuit in a matroid. Let M be a matroid defined by an independence oracle on some ground set U, and let $A_1, \ldots, A_n$ be given (not necessarily disjoint) subsets of U. We define a *generalized circuit* as a minimal subset X of $S = \{1, \ldots, n\}$ such that $\bigcup_{i \in X} A_i$ is a dependent set in M.

PROPOSITION 4 (see [3]). *Enumerating all generalized circuits for the cocycle matroid of a graph is NP-hard even when $A_1, \ldots, A_n$ are disjoint sets of edges of size 2 each.*

Proposition 4 also holds for cycle matroids. In addition, by matroid duality, Proposition 3 shows that it may be NP-hard to enumerate all *generalized hyperplanes* of a matroid M, i.e., all those maximal subsets X of $S = \{1, \ldots, n\}$ for which $Span(\cup_{i \in X} A_i) \neq S$.

In the next section we consider *generalized bases* ($\equiv$ *minimal spanning sets*) and argue that their enumeration is an easier task even when they are defined for two or more matroids.

5. Generalized spanning and packing in matroids and systems of polymatroid inequalities.

5.1. Minimal spanning sets. Given a matroid M on ground set U and subsets $A_1, \dots, A_n$ of U, we define a *generalized base* as a minimal subset X of $S = \{1, \dots, n\}$ for which the sets $A_i,\ i \in X$ span the matroid, that is,

$$r\Big(\bigcup_{i\in X} A_i\Big) \geq r(M). \tag{7}$$

Note that in case $|U| = n$ and the sets A_i are the n disjoint singletons of U, we obtain the standard definition of a base of M. On the other hand, if A_i are subsets of some fixed base B, then

$$r\Big(\bigcup_{i\in X} A_i\Big) = \Big|\bigcup_{i\in X} A_i\Big|,$$

and hence each generalized base is a minimal set cover of B. Another special case will be considered in section 5.3.1 below.

It will be convenient to further extend the definition of generalized bases. First, we can replace the right-hand side of inequality (7) by a given integer threshold $t \in \{1, \dots, r(M)\}$. This is equivalent to replacing M by the truncated matroid with the rank function $r_t(\cdot) = \min\{r(\cdot), t\}$ and leads us to the notion of a *minimal t-spanning set*, that is, a minimal set $X \subseteq \{1, \dots, n\}$ for which

$$r\Big(\bigcup_{i\in X} A_i\Big) \geq t.$$

Naturally, for $t = r(M)$ we return to the definition of generalized bases.

Second, we may consider a number m of matroids $M_1, \dots, M_m$ defined by independence oracles on ground sets $U_1, \dots, U_m$. Suppose that for each of the matroids M_j we are given a collection of n sets $A_{j1}, \dots, A_{jn} \subseteq U_j$ along with an integer threshold $t_j \in \{1, \dots, r(M_i)\}$, and consider the family $\mathcal{F}$ of all minimal solutions $X \subseteq S = \{1, \dots, n\}$ to the system of m inequalities

$$r_j\Big(\bigcup_{i\in X} A_{ji}\Big) \geq t_j, \quad j = 1, \dots, m. \tag{8}$$

Let us recall that an integer-valued set-function $f : 2^S \mapsto \mathbb{Z}_+$ is called *polymatroid* if it is monotone, submodular, with minimum 0, i.e.,

(i) $X \subseteq Y \subseteq S \implies f(X) \leq f(Y)$,

(ii) $f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)$ for all $X, Y \subseteq S$, and

(iii) $f(\emptyset) = 0$.

For example, the rank function of any matroid is *polymatroid*. It is also easy to see that if $A_1, \dots, A_n$ are arbitrary subsets of the ground set U of a matroid with the rank function r, then the function $f(X) = r(\cup_{i\in X} A_i)$ is polymatroid. This function $f(X) = r(\cup_{i\in X} A_i)$ is called the *parallel extension* of r *with respect to* $A_1, \dots, A_n$; see, e.g., [17]. It is known [12] that any polymatroid function f can be obtained in this way from some matroid.

5.2. Systems of polymatroid inequalities. Returning to the minimal spanning subfamilies for (7) or, more generally, to the family $\mathcal{F}$ of all minimal feasible solutions for (8), we conclude that $\mathcal{F}$ is the family of all minimal feasible solutions $X \subseteq S = \{1, \dots, n\}$ to the system of polymatroid inequalities

$$f_j(X) \geq t_j, \quad j = 1, \dots, m, \tag{9}$$

where $f_j(X) = r_j(\bigcup_{i \in X} A_{ji})$. Since we assume that each of the input matroids is given by an independence oracle, each of the functions f_j can be evaluated at any set $X \subseteq S$ in polynomial oracle time. As shown in [4], given a system of polymatroid inequalities (9) defined by a polynomial-time evaluation oracle, we can compute any given number $k \leq |\mathcal{F}|$ of minimal solutions $X \in \mathcal{F}$ to (9) in time $N^{o(\log N)}$, where $N = m(nk)^{\log \max\{t_1, \dots, t_m\}}$. Since $t_j \leq |U_j|, \quad j = 1, \dots, m$, we thus conclude that all minimal solutions to (8) can be enumerated in *incremental quasi-polynomial* time.

THEOREM 3 (see [4]). *Given m matroids $M_1, \dots, M_m$ defined by independence oracles on ground sets $U_1, \dots, U_m$ and given a collection of n sets $A_{j1}, \dots, A_{jn} \subseteq U_j$ along with an integer threshold $t_j \in \{1, \dots, r(M_j)\}$ for each of the matroids, the set $\mathcal{F}$ of minimal solutions $X \subseteq \{1, \dots, n\}$ to the system of generalized rank inequalities* (8) *can be computed in incremental quasi-polynomial time; i.e., $k \leq |\mathcal{F}|$ elements of $\mathcal{F}$ can be produced in $2^{\mathrm{polylog}(K)}$ time, where $K = \max\{k, n, m, |U_1|, \dots, |U_m|\}$.*

Theorem 3 clearly indicates that for matroids defined by polynomial-time independence oracles, the enumeration of all generalized bases or, more generally, all minimal solutions for (8) is very unlikely to be NP-hard.

A function $f : 2^S \to \Re$ is called *ℓ-smooth* if $|f(X \cup \{i\}) - f(X)| \leq \ell$ for all $X \subseteq S$ and $i \in S \setminus X$. Obviously, the rank function r of any matroid is 1-smooth, while the parallel extension of r with respect to given sets $A_1, \dots, A_n$ is $\max\{|A_1|, \dots, |A_n|\}$-smooth. When the number of polymatroid inequalities in (9) is fixed and each of these inequalities is ℓ-smooth for some fixed ℓ, Theorem 3 can be strengthened as follows.

THEOREM 4. *When* $\max\{m, |A_{11}|, \dots, |A_{nm}|\} \leq \text{const}$ *all minimal solutions to* (9) *can be enumerated with polynomial delay.*

Proof. Let $\ell \overset{\text{def}}{=} \max\{|A_{11}|, \dots, |A_{nm}|\}$, and note that the functions $f_1, \dots, f_m$ are ℓ-smooth. Denote by $\mathcal{F}$ the family of minimal feasible sets for (9). Then the elements of $\mathcal{F}$ can be enumerated with polynomial delay by traversing the *strongly connected* directed supergraph $\mathcal{G} = (\mathcal{F}, \mathcal{E})$, in which a pair of vertices (X, X') forms an edge in $\mathcal{E}$ if and only if X' can be obtained from X by the following process:

1. Let e be an element of X such that $S \setminus e$ satisfies the system (9). Delete e from X.
2. Add a set Z of at most $m\ell$ elements from $S \setminus X$ to restore the feasibility of $(X \setminus e) \cup Z$.
3. Lexicographically delete some elements Y from $X \setminus e$ to guarantee the minimality of $X' = (X \setminus (Y \cup e)) \cup Z$.

Note that for bounded ℓ and m, the out-degree of each vertex of $\mathcal{G}$ is polynomially bounded by $n^{m\ell+1}$. Furthermore, the strong connectivity of $\mathcal{G}$ can be proved as follows. Given two vertices $X_0, X_k \in \mathcal{F}$ of $\mathcal{G}$, there exists a sequence $\{X_1, \dots, X_{k-1}\}$ of elements of $\mathcal{F}$ such that for all $r = 1, \dots, k$, X_r is obtained from X_{r-1} by deleting an element $e_r \in X_{r-1} \setminus X_k$ (thus making $X_{r-1} \setminus \{e_r\}$ infeasible), adding some minimal subset of elements $Z_r \subseteq X_k \setminus X_{r-1}$ to obtain a feasible set $X_{r-1} \setminus \{e_r\} \cup Z_r$, and finally, reducing lexicographically the resulting set to a minimal feasible set $X_r \subseteq X_{r-1} \cup Z_r \setminus \{e_r\}$. Note that, for $r = 1, \dots, k$, $|X_r \setminus X_k| < |X_{r-1} \setminus X_k|$, and therefore $k \leq |X_0 \setminus X_k|$. Thus it is enough to show that $(X_{r-1}, X_r) \in \mathcal{E}$, i.e., there exists a subset $Z_r \subseteq X_k \setminus X_{r-1}$ such that

$$|Z_r| \le m\ell, \tag{10}$$

for $r = 1, \dots, k-1$. To prove this, fix an $r \in \{1, \dots, k-1\}$, and let, for simplicity of notation, $X = X_{r-1}$, $e = e_r$, and $Z = Z_r$. We begin by initializing $Z \leftarrow \emptyset$. As long as $f_j((X \setminus \{e\}) \cup Z) < t_j$ for some $j = 1, \dots, m$, there must exist an element $x \in X_k \setminus ((X \setminus \{e\}) \cup Z)$ such that $f_j((X \setminus \{e\}) \cup Z \cup \{x\}) > f_j((X \setminus \{e\}) \cup Z)$. This follows from the submodularity of f_j and the fact that

$$f_j((X \setminus \{e\}) \cup Z) < t_j \le f_j(X_k) \le f_j((X \setminus \{e\}) \cup Z \cup X_k).$$

We then add this element x to Z, i.e., set $Z \leftarrow Z \cup \{x\}$, which increases the value of $f_j((X \setminus \{e\}) \cup Z)$ by at least 1. By the ℓ-smoothness of f_j, for $j = 1, \dots, m$, we have $f_j(X \setminus \{e\}) \ge t_j - \ell$, and therefore we conclude that the number of elements that we need to include in Z before $(X \setminus \{e\}) \cup Z$ satisfies the system (9) is at most $m\ell$. The theorem follows. □

Remark. Let us note that in reducing the set $X_{r-1} \setminus \{e_r\} \cup Z_r$ to a minimal feasible set X_r for (9), in any possible way, we need only to delete at most $m(\ell(m\ell+1)-1)-1 = O(m^2\ell^2)$ elements, i.e., $X_r = X_{r-1} \cup Z_r \setminus (\{e\} \cup Y_r)$, where

$$|Y_r| \le m(\ell(m\ell+1)-1)-1 \le 2\ell^2 m^2. \tag{11}$$

To see this, let, as before, $X = X_{r-1}$, $e = e_r$, $Z = Z_r$, $Y = Y_r$, and $X' = X_r$. By the minimality of $X \in \mathcal{F}$, for each $x \in X$, there is a $j \in [m] \overset{\text{def}}{=\!=} \{1, \dots, m\}$ such that $f_j(X \setminus \{x\}) \le t_j - 1$. Let $I \overset{\text{def}}{=} \{j \in [m] \ : \ f_j(X \setminus \{x\}) \le t_j - 1, \text{ for some } x \in X\}$. Since $X \setminus \{e\}$ is infeasible, there exists a $j' \in I$ such that $f_{j'}(X \setminus \{e\}) \le t_{j'} - 1$. Note that for each $j \in I$, we have

$$f_j(X) \le t_j + \ell - 1 \tag{12}$$

by the minimality of X and the ℓ-smoothness of f_j. For $j \in I$, define $Y_j = \{x \in Y \cup \{e\} \mid f_j(X \setminus \{x\}) \le t_j - 1\}$. Since $X' = X \cup Z \setminus (Y \cup \{e\}) \subseteq X \cup Z \setminus Y_j$ for all $j \in I$, we get

$$f_j(X') \le f_j(X \setminus Y_j) + \ell|Z| \quad \text{for all } j \in I. \tag{13}$$

It follows, furthermore, by the submodularity of f_j and minimality of X that

$$f_j(X) - f_j(X \setminus Y_j) \ge |Y_j| \quad \text{for all } j \in I. \tag{14}$$

But $f_j(X') \ge t_j$ for all $j \in I$, which together with (12), (13), and (14) gives

$$|Y \cup \{e\}| \le \sum_{j \in I} |Y_j| \le \sum_{j \in I} (f_j(X) - f_j(X \setminus Y_j)) \le |I|(\ell(|Z|+1)-1).$$

Now (11) follows from (10).

Let us further remark that, for a single polymatroid inequality, the bounds (10) and (11) can be tight to within multiplicative constants. To see this, consider for instance the grid $V = \{0, 1, \dots, \ell\} \times \{0, 1, \dots, \ell\}$, where ℓ is a positive integer. Let $n = (\ell+1)^2$, and consider the following collection of n edge sets on V:

$$\begin{aligned} E_{ji} &= \{\{(j,0),(j,i)\}\} \quad \text{for } i = 1, \dots, \ell \text{ and } j = 0, 1, \dots, \ell, \\ E_* &= \{\{(0,0),(j,0)\} \mid j = 1, \dots, \ell\}, \\ E_j &= \{\{(j,1),(j,i)\} \mid i = 2, \dots, \ell\} \cup \{\{(j,1),(j+1,\ell)\}\} \\ &\qquad\qquad\qquad\qquad\qquad\qquad \text{for } j = 0, 1, \dots, \ell-1. \end{aligned}$$

Let $G_1, \ldots, G_n$ be the graphs corresponding to the above edge sets on V. For $X \subseteq \{1, \ldots, n\}$, define

$$f(X) = |V| - (\text{number of connected components of the graph } \textstyle\bigcup_{i \in X} G_i).$$

Then f is polymatroid (see Example 5.3.1 below) and ℓ-smooth. Let $t = |V| - 1$, and consider the polymatroid inequality

$$f(X) \geq t. \tag{15}$$

Then the set $X = \{E_{ji} \mid i = 1, \ldots, \ell,\ \ j = 0, 1, \ldots, \ell\} \cup \{E_*\}$ is minimal feasible for (15). Now if we drop E_* from X, we need to add all the ℓ sets E_j for $j = 0, 1, \ldots, \ell-1$ to restore feasibility. But then all the $\ell(\ell-1)$ sets E_{ji} for $i = 2, 3, \ldots, \ell$ and $j = 0, 1, \ldots, \ell - 1$ can be dropped to get back to minimal feasibility. □

As a corollary of (11) and Theorem 4, we obtain the following claim.

COROLLARY 2. *Given a nonempty collection $\mathcal{X}$ of minimal feasible solutions for* (9)*, the problem of finding a new minimal feasible solution $X \notin \mathcal{X}$ is in NC, provided that* $\max\{m, |A_{11}|, \ldots, |A_{nm}|\} \leq \text{const}$.

In contrast, it was proven in [14] that the parallel complexity of finding a single maximal independent set in a matroid, defined by an independence oracle on a ground set S, using p processors, is lower-bounded (in a certain decision tree model) by $\Omega((|S|/\log(|S|p))^{1/3})$.

5.3. Spanning and packing in matroids. In the remainder of this paper we briefly discuss some applications of Theorems 3 and 4.

5.3.1. Minimal *t*-spanning sets and connectivity ensuring collections of graphs. Theorems 3 and 4 imply that, given a matroid M on ground set U and an integer threshold t, the family

$$\mathcal{F}(\mathcal{A}, t) = \{X \mid X \text{ minimal subset of } \{1, \ldots, n\} \text{ such that } r(\cup_{i \in X} A_i) \geq t\}$$

of all minimal t-spanning sets can be enumerated in incremental quasi-polynomial time for any given collection $\mathcal{A}$ of sets $A_1, \ldots, A_n \subseteq U$ and with polynomial delay when the sizes of $A_1, \ldots, A_n$ are bounded. In particular, this result applies to generalized bases, i.e., minimal subfamilies of $A_1, \ldots, A_n$ that span the entire matroid:

$$Span\left\langle \bigcup_{i \in X} A_i \right\rangle = U.$$

As an application, let $A_1, \ldots, A_n \subseteq V \times V$ be a family of edge sets of undirected graphs on common vertex set V, and let $\mathcal{F}$ be all those minimal subfamilies X of $\{1, \ldots, n\}$ for which the graph $G(X) = (V, \bigcup_{i \in X} A_i)$ is connected (or has at most t connected components, where t is a given threshold). Then all elements of $\mathcal{F}$ can be enumerated in incremental polynomial time or with polynomial delay for $\max\{|A_1|, \ldots, |A_n|\} \leq$ const. This result generalizes the well-known fact that all spanning trees for a graph can be enumerated efficiently. Interestingly, enumerating all minimal collections of $A_1, \ldots, A_n$ *connecting two given vertices* $a, a' \in V$ turns out to be NP-hard already when the input sets of edges $A_1, \ldots, A_n$ are pairwise disjoint and contain at most 2 edges each; see [11]. In other words, given n disjoint sets $A_1, \ldots, A_n \subseteq U$ of size 2 in a (graphic) matroid M, it is NP-hard to enumerate all minimal subfamilies $X \subseteq \{1, \ldots, n\}$ which span a *given flat* A of the matroid

$$A \subseteq Span\left\langle \bigcup_{i \in X} A_i \right\rangle,$$

even when A is a line, i.e., $r(A) = 1$. In addition, Proposition 4 shows that generating the family of all generalized hyperplanes

$$\mathcal{H}(\mathcal{A}) = \left\{ X \;\middle|\; \begin{array}{c} X \text{ is a maximal subset of } \{1, \dots, n\} \\ \text{such that } r(\cup_{i \in X} A_i) \leq r(M) - 1 \end{array} \right\}$$

is also NP-hard already for graphic matroids and $|A_1| = \cdots = |A_n| = 2$.

5.3.2. Bar-and-joint structures. The following example is taken from [17]. "Let B be a bar-and-joint structure, i.e., a graph $G = (V, E)$ whose nodes are points of the Euclidean 3-space, and whose edges are rigid bars attached to the nodes by flexible joints. For each $X \subseteq V$, let $f(X)$ denote the *degree of freedom* of the subset X, i.e., the dimension of the infinitesimal motions of all nodes in X which extend to an infinitesimal motion of all nodes that is compatible with the given bars. Then $f(\emptyset) = 0$, $f(\{x\}) = 3$ for every $x \in V(G)$ and $f(\{x, y\}) = 5$ or 6 depending on whether or not the whole structure forces x and y to stay at the same distance, etc. It follows from the elements of the theory of rigid bar-and-joint structures that f is a submodular set-function on the subsets of $V(G)$. See [5]."

Moreover, it is easy to see that function f is polymatroid. Hence, it follows from Theorem 3 that, given a positive integer threshold t, we can generate all minimal families of nodes whose degree of freedom is at least t in incremental quasi-polynomial time. Of course, this result can be generalized to parallel extensions of f and to families of bar-and-joint structures with different thresholds.

5.3.3. Maximal independent sets in m matroids. An important special case of Theorem 3 is when all matroids are defined on the same ground set $U = U_1 = \cdots = U_n = \{1, \dots, n\}$, and $A_{ji} = \{i\}$ for all $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, r\}$. For this case, defining $t_j = |U| - r_j(U)$ and writing the system (8) for the dual matroids $M_1^*, \dots, M_m^*$,

$$r_j^*(X) = r_j(U \setminus X) + |X| - r_j(U) \geq |U| - r_j(U), \quad j = 1, \dots, m, \tag{16}$$

we notice that the complement $Y = U \setminus X$ to each minimal feasible solution X to (16) is a maximal set Y independent in all m matroids

$$r_j(Y) \geq |Y|, \quad j = 1, \dots, m,$$

and vice versa. Thus, the set of all minimal feasible solutions to (16) can be identified with the family

$$cI(M_1, \dots, M_m) = \{Y \mid Y \text{ is a maximal set independent in } M_1, \dots, M_m\}.$$

The complexity of enumerating $\mathcal{I}(M_1, \dots, M_m)$ was asked in 1980 by Lawler, Lenstra, and Rinnooy Kan [15], who gave an algorithm running in exponential time $O(n^{m+2})$ per each generated element. Theorem 3 indicates that in fact this enumeration problem (called *matroid intersections* in [15]) can be solved in incremental quasi-polynomial time.

THEOREM 5 (see [2]). *Given m matroids $M_1, \dots, M_m$ on common ground set U, we can enumerate $k \leq |\mathcal{I}(M_1, \dots, M_m)|$ maximal independent sets in $N^{o(\log N)}$ time and poly(N) calls to the independence oracles, where $N = \max\{m, k, |U|\}$.*

5.3.4. Minimal transversals of bounded-degree hypergraphs. Let $\mathcal{H} \subseteq 2^{\{1,\dots,n\}}$ be a hypergraph on n vertices. A minimal transversal of $\mathcal{H}$ is a subset of vertices, minimal with the property that it intersects every hyperedge of $\mathcal{H}$. The problem of finding all minimal transversals of a given hypergraph $\mathcal{H}$ is known as the hypergraph transversal problem (see, e.g., [8, 10]) and is equivalent to the hypergraph dualization problem mentioned in the introduction. For $X \subseteq \{1,\dots,n\}$, define the function

$$f_{\mathcal{H}}(X) = |\{H \in \mathcal{H} \mid H \cap X \neq \emptyset\}|.$$

Then f is polymatroid; moreover, it is ℓ-smooth whenever the hypergraph $\mathcal{H}$ has maximum degree ℓ, i.e.,

$$\deg(\mathcal{H}) = \max_{i \in V} |\{H \in \mathcal{H} \mid i \in H\}| \leq \ell.$$

Since the dualization of $\mathcal{H}$ is equivalent to the enumeration of all minimal solutions to the inequality $f_{\mathcal{H}}(X) \geq |\mathcal{H}|$, we conclude by Theorem 4 that bounded-degree hypergraphs can be dualized with polynomial delay (see [6] and [9] for alternative proofs of this result).

5.3.5. Packing flats in a matroid. In our final example, given a matroid M on ground set U, subsets $A_1,\dots,A_n \subseteq U$, and an integer threshold t, we consider the inequality

$$\sum_{i \in X} r(A_i) - r\left(\bigcup_{i \in X} A_i\right) \leq t, \tag{17}$$

where $X \subseteq \{1,\dots,n\}$ and r is the rank function of M. It is easy to see that the left-hand side of (17) monotonically increases with X. We call maximal solutions to (17) *t-packings* of $A_1,\dots,A_n$ in M. When $t = 0$ and $r(X) = |X|$, this definition leads to the usual notion of set packings, i.e., maximal pairwise disjoint subfamilies of $A_1,\dots,A_n$. When $t = 0$ and $A_1,\dots,A_n$ are some flats, for instance, some subspaces of a linear space, each packing is a maximal mutually transversal subfamily of flats. This is because for $t = 0$ the packing inequality (17) is equivalent to $|X|$ transversality conditions: for each $i \in X$ the flat A_i must be transversal to the flat generated by all other flats, i.e., $r(A_i \cup L_i) = r(A_i) + r(L_i)$, where $L_i = \mathit{Span}\langle A_j \mid j \in X \setminus i\rangle$. In particular, when M is the cycle matroid of a graph G and $A_1,\dots,A_n$ are subgraphs of G, then for $t = 0$ each maximal packing corresponds to maximal pairwise *edge-disjoint* subfamilies of graphs the union of which does not contain any new cycle (i.e., a cycle that does not belong to one of the subgraphs $A_1,\dots,A_n$).

Rewriting inequality (17) as

$$\sum_{i \in Y} r(A_i) + r\left(\bigcup_{i \notin Y} A_i\right) \geq \sum_{i=1}^{n} r(A_i) - t,$$

where $Y = \{1,2,\dots,n\} \setminus X$, we can note that the maximal packings in (17) are in one-to-one correspondence with the minimal feasible sets of a polymatroid inequality. Thus we arrive at the following result.

COROLLARY 3. *Given a matroid M on ground set U, and given subsets $A_1,\dots,A_n \subseteq U$, all maximal t-packings of $A_1,\dots,A_n$ can be enumerated in incremental quasi-polynomial time for any given integer threshold t.*

In particular, given a family of acyclic graphs on the same vertex set V, all maximal *edge-disjoint* subfamilies, the union of which is still acyclic, can be enumerated incrementally in quasi-polynomial time. In contrast to this result, a seemingly very similar problem is NP-hard.

PROPOSITION 5. *Given a family of acyclic graphs on the same vertex set V, it is NP-hard to enumerate all maximal (not necessarily edge-disjoint) subfamilies, the union of which is still acyclic.*

Proof. We use the following reduction from the CNF satisfiability problem. Given a conjunctive normal form $\phi(x_1, \dots, x_n) = C_1 \wedge \cdots \wedge C_m$, where each C_j is a disjunction of some literals in $\{x_1, \bar{x}_1, \dots, x_n, \bar{x}_n\}$, our construction is composed of the union G of $n+m$ vertex disjoint cycles $Y_1, \dots, Y_n, Y'_1, \dots, Y'_m$, where $|Y_i| = n+2$ for $i = 1, \dots, n$ and $|Y'_j| = |C_j|+n$ for $j = 1, \dots, m$. Each edge of G is labeled by one of the sets $\{x_i\}$, $\{\bar{x}_i\}$, or $\{x_i, \bar{x}_i\}$ for some $i \in \{1, \dots, n\}$. Each cycle Y_i, for $i = 1, \dots, n$, corresponds to a binary variable x_i, and its $n+2$ edges are labeled by $\{x_1, \bar{x}_1\}, \dots, \{x_n, \bar{x}_n\}$, $\{x_i\}$, and $\{\bar{x}_i\}$, respectively. Similarly, each cycle Y'_j, for $j = 1, \dots, m$, corresponds to a clause C_j, and its edges are labeled by $\{x_1, \bar{x}_1\}, \dots, \{x_n, \bar{x}_n\}$, $\{l_1\}, \dots, \{l_{k_j}\}$, where $l_1, \dots, l_{k_j}$ are the literals appearing in C_j. Finally, we define $2n$ acyclic subgraphs of G with edge sets $X_1, \overline{X}_1, \dots, X_n, \overline{X}_n$, where $X_i = \{e \in E(G) \mid x_i \in \text{label}(e)\}$ and $\overline{X}_i = \{e \in E(G) \mid \bar{x}_i \in \text{label}(e)\}$. Note that, for any $j \in \{1, \dots, n\}$, the family $\{X_i \mid i \neq j\} \cup \{\overline{X}_i \mid i \neq j\}$ is maximal with the property that the union graph is acyclic. Thus any other maximal family whose union is acyclic must contain either X_i or $\bar{X}_i$ for all $i = 1, \dots, n$. Furthermore, any such family cannot contain both X_i and $\overline{X}_i$ for some $i \in \{1, \dots, n\}$ since otherwise we get the cycle Y_i in the union. We conclude, therefore, that there exists a new maximal union-acyclic subfamily if and only if there is a subfamily containing exactly one of the sets $X_i, \overline{X}_i$ for all $i = 1, \dots, n$ such that all the cycles $Y'_1, \dots, Y'_m$ are broken in the union graph. The latter condition is further equivalent with the condition that CNF ϕ is satisfiable. □

As a final remark, we note that the family of sets described in Proposition 5 is the family of maximal feasible solutions of the inequality

$$F(X) \overset{\text{def}}{=} \left|\bigcup_{i \in X} A_i\right| - r\left(\bigcup_{i \in X} A_i\right) \leq 0$$

over $X \subseteq \{1, \dots, n\}$, where $r(\cdot)$ is the rank function of the cycle matroid of the graph G. In the special case, when $n = |U|$ and the sets A_i are the n disjoint singletons of U, the function $F(X)$ reduces to the *copolymatroid* function $f(X) \overset{\text{def}}{=} |X| - r(X)$ (that is, $f(U) - f(U \setminus X)$ is polymatroid). More generally, it follows that, for any integer t, the maximal feasible solutions of the inequality $f(X) \leq t$ can be enumerated with polynomial delay by Theorem 4, since those are in a one-to-one correspondence with the minimal feasible solutions of the 1-smooth polymatroid inequality $n - t - r(U) \leq |Y| + r(U \setminus Y) - r(U)$. Analogous argument shows that the minimal feasible solutions of the inequality $f(X) \geq t$ are the complements of flats of rank $t - n - r(U) - 1$ in the dual matroid and hence can be enumerated in incremental polynomial time by Proposition 1. In contrast, Propositions 4 and 5 state that the analogous problems for the function $F(X)$ (which is a parallel extension of $f(X)$) are NP-hard. Thus, while the parallel extension of a polymatroid function is polymatroid, the parallel extension of a copolymatroid function is *neither* polymatroid *nor* copolymatroid in general, and thus the corresponding generation problems may be NP-hard.

Acknowledgment. The authors thank the referees for helpful remarks.

REFERENCES

[1] R. Bixby and W. Cunningham, *Matroid Optimization and Algorithms*, in Handbook of Combinatorics, R. L. Graham, M. Grötschel, and L. Lovász, eds., North–Holland, Amsterdam, 1995, pp. 550–609.

[2] E. Boros, K. Elbassioni, V. Gurvich, and L. Khachiyan, *Matroid intersections, polymatroid inequalities, and related problems*, in Proceedings of the 27th International Symposium on Mathematical Foundations of Computer Science, MFCS, Warsaw, 2002, pp. 143–154.

[3] E. Boros, K. Elbassioni, V. Gurvich, and L. Khachiyan, *Generating dual-bounded hypergraphs*, Optim. Methods Softw., 17 (2002), pp. 749–781.

[4] E. Boros, K. Elbassioni, V. Gurvich, and L. Khachiyan, *Extending the Balas-Yu inequality on the number of maximal independent sets of graphs to hypergraphs and lattice products with applications*, Math. Program. Ser. B, 98 (2003), pp. 355–368.

[5] H. Crapo, *Structural rigidity*, Struct. Topol. 1 (1979), pp. 26–45.

[6] C. Domingo, N. Mishra, and L. Pitt, *Efficient read-k monotone CNF/DNF dualization using a learning with membership queries approach*, Machine Learning, 37 (1999), pp. 89–110.

[7] J. Edmonds, *Submodular functions, matroids, and certain polyhedra*, in Combinatorial Structures and Their Applications, R. Guy, H. Hanani, N. Sauer, and J. Shönheim, eds., Gordon and Breach, New York, 1970, pp. 69–87.

[8] T. Eiter and G. Gottlob, *Identifying the minimal transversals of a hypergraph and related problems*, SIAM J. Comput., 24 (1995), pp. 1278–1304.

[9] T. Eiter, G. Gottlob, and K. Makino, *New results on monotone dualization and generating hypergraph transversals*, in Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC), Montréal, Quebec, Canada, 2002, pp. 14–22.

[10] M. L. Fredman and L. Khachiyan, *On the complexity of dualization of monotone disjunctive normal forms*, J. Algorithms, 21 (1996), pp. 618–628.

[11] V. Gurvich and L. Khachiyan, *On generating the irredundant conjunctive and disjunctive normal forms of monotone Boolean functions*, Discrete Appl. Math., 96-97 (1999), pp. 363–373.

[12] T. Helgason, *Aspects of the theory of hypermatroids*, in Hypergraph Seminar, Lecture Notes in Math. 411, C. Berge and D. K. Ray-Chaudhuri, eds., Springer, New York, 1975, pp. 191–214.

[13] R. M. Karp, *On the complexity of combinatorial problems*, Networks, 5 (1975), pp. 45–68.

[14] R. Karp, E. Upfal, and A. Wigderson, *The complexity of parallel search*, J. Comput. System Sci., 36 (1988), pp. 225–253.

[15] E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, *Generating all maximal independent sets: NP-hardness and polynomial-time algorithms*, SIAM J. Comput., 9 (1980), pp. 558–565.

[16] A. Lehman, *A solution of the Shannon switching game*, J. Soc. Indust. Appl. Math., 12 (1964), pp. 687–725.

[17] L. Lovász, *Submodular functions and convexity*, in Mathematical Programming: The State of the Art, Bonn 1982, Springer, New York, 1983, pp. 235–257.

[18] J. S. Provan and D. R. Shier, *A paradigm for listing (s, t)-cuts in graphs*, Algorithmica, 15 (1996), pp. 357–372.

[19] R. C. Read and R. E. Tarjan, *Bounds on backtrack algorithms for listing cycles, paths, and spanning trees*, Networks, 5 (1975), pp. 237–252.

[20] N. Robertson and P. D. Seymour, *Graph minors.* XIII. *The disjoint path problem*, J. Combin. Theory Ser. B, 63 (1995), pp. 65–110.

[21] P. D. Seymour, *A note on hyperplane generation*, J. Combin. Theory Ser. B, 61 (1994), pp. 88–91.

[22] V. Vazirani, *Approximation Algorithms*, Springer, Berlin, 2001.

[23] D. J. A. Welsh, *Matroid Theory*, Academic Press, London, New York, San Francisco, 1976.

SIAM J. Discrete Math.
Vol. 19, No. 4, pp. 985–1003

LABELLING CAYLEY GRAPHS ON ABELIAN GROUPS[*]

SANMING ZHOU[†]

In memory of Xin-Bang Yan who turned on my interest in mathematics

Abstract. For given integers $j \geq k \geq 1$, an $L(j,k)$-labelling of a graph Γ is an assignment of labels—nonnegative integers—to the vertices of Γ such that adjacent vertices receive labels that differ by at least j, and vertices distance two apart receive labels that differ by at least k. The span of such a labelling is the difference between the largest and the smallest labels used, and the minimum span over all $L(j,k)$-labellings of Γ is denoted by $\lambda_{j,k}(\Gamma)$. The minimum number of labels needed in an $L(j,k)$-labelling of Γ is independent of j and k, and is denoted by $\mu(\Gamma)$. In this paper we introduce a general approach to $L(j,k)$-labelling Cayley graphs Γ over Abelian groups and deriving upper bounds for $\lambda_{j,k}(\Gamma)$ and $\mu(\Gamma)$. Using this approach we obtain upper bounds on $\lambda_{j,k}(\Gamma)$ and $\mu(\Gamma)$ for graphs Γ admitting a vertex-transitive Abelian group of automorphisms. Hypercubes Q_d are examples of such graphs, and as consequences we obtain upper bounds for $\lambda_{j,k}(Q_d)$ and $\mu(Q_d)$. We also obtain the exact values of $\lambda_{j,k}(\Gamma)$ $(2k \geq j \geq k)$ and $\mu(\Gamma)$ for some Hamming graphs Γ. The result shows that, under certain arithmetic conditions, these two invariants rely only on k and the orders of the two largest complete graph factors of the Hamming graph.

Key words. channel assignment, $L(j,k)$-labelling, $\lambda_{j,k}$-number, λ-number, radio chromatic number, Cayley graph, hypercube, Hamming graph

AMS subject classification. 05C78

DOI. 10.1137/S0895480102404458

1. Introduction.

In the *channel assignment problem* [13] one wishes to assign channels to the transmitters in a radio communication system such that interference is avoided as much as possible. For this purpose various constraints have been proposed [13, 22] to put on the channel separations between pairs of transmitters within certain distance. It is suggested [11] that "close" transmitters be assigned channels at least k apart, and "very close" transmitters be assigned channels at least j apart, where j and k are given integers with $j \geq k \geq 1$. Since bandwidth is a limited resource, a major concern is to minimize the span of channels used. Taking channels as nonnegative integers, this problem can be formulated as a labelling problem [7, 11] for the corresponding interference graph. More explicitly, for a graph $\Gamma = (V(\Gamma), E(\Gamma))$ with vertex set $V(\Gamma)$ and edge set $E(\Gamma)$, a mapping f from $V(\Gamma)$ to $\mathbb{Z}^+ = \{0, 1, 2, \ldots\}$ is called [7, 11] an *$L(j,k)$-labelling* of Γ if, for any $u, v \in V(\Gamma)$,

$$d_\Gamma(u,v) = 1 \Rightarrow |f(u) - f(v)| \geq j$$

and

$$d_\Gamma(u,v) = 2 \Rightarrow |f(u) - f(v)| \geq k,$$

where $d_\Gamma(u,v)$ is the distance in Γ between u and v. The integer $f(u)$ is called the *label* of u under f, and $\mathrm{sp}(\Gamma; f) = \max_{u \in V(\Gamma)} f(u) - \min_{v \in V(\Gamma)} f(v)$ the *span* of

[*] Received by the editors March 25, 2002; accepted for publication (in revised form) June 24, 2005; published electronically January 26, 2006. This research was supported by Discovery Project grants DP0344803 and DP0558677 from the Australian Research Council and by a Melbourne Early Career Researcher Grant from The University of Melbourne.

http://www.siam.org/journals/sidma/19-4/40445.html

[†] Department of Mathematics and Statistics, The University of Melbourne, Parkville, VIC 3010, Australia (smzhou@ms.unimelb.edu.au).

f. Without loss of generality *we will always assume that the smallest label used by an $L(j,k)$-labelling is* 0. With this convention the span of f is equal to the largest label used by f, that is, $\mathrm{sp}(\Gamma; f) = \max_{u \in V(\Gamma)} f(u)$. The *$\lambda_{j,k}$-number* of Γ, denoted by $\lambda_{j,k}(\Gamma)$, is defined [7, 11] to be the minimum span of all $L(j,k)$-labellings of Γ. Usually, $\lambda_{2,1}(\Gamma)$ is called [11] the *λ-number* of Γ and is denoted by $\lambda(\Gamma)$.

A relevant invariant is the minimum number $\mu_{j,k}(\Gamma)$ of labels needed in an $L(j,k)$-labelling of Γ. This invariant is actually independent of choice of j and k [26], that is, for any $j \geq k \geq 1$,

$$\mu_{j,k}(\Gamma) = \mu_{1,1}(\Gamma). \tag{1}$$

In fact, since $j \geq k \geq 1$, any $L(j,k)$-labelling of Γ is an $L(1,1)$-labelling of Γ, and hence $\mu_{1,1}(\Gamma) \leq \mu_{j,k}(\Gamma)$. On the other hand, for any $L(1,1)$-labelling of Γ using $\mu_{1,1}(\Gamma)$ labels, by multiplying the label of each vertex by j we obtain an $L(j,k)$-labelling of Γ which uses $\mu_{1,1}(\Gamma)$ labels. Therefore, we have $\mu_{j,k}(\Gamma) \leq \mu_{1,1}(\Gamma)$ and (1) follows. In the following we will denote $\mu_{1,1}(\Gamma)$ by $\mu(\Gamma)$. Thus, in view of (1), $\mu_{j,k}(\Gamma)$ is equal to $\mu(\Gamma)$ for any $j \geq k \geq 1$. An $L(j,k)$-labelling of Γ is said to be *optimal for $\lambda_{j,k}$* if its span is $\lambda_{j,k}(\Gamma)$, and *optimal for μ* if it uses $\mu(\Gamma)$ distinct labels. In particular, an $L(2,1)$-labelling of Γ is *optimal for λ* if its span is $\lambda(\Gamma)$. Note that an $L(j,k)$-labelling of Γ which is optimal for $\lambda_{j,k}$ is not necessarily optimal for μ and vice versa.

The $L(j,k)$-labelling problem, in particular in the $L(2,1)$ case, has been studied extensively in the past more than one decade; see [2, 3, 4, 6, 7, 8, 9, 10, 11, 19, 22, 24, 29, 30] for examples. The $L(2,1)$-labelling problem was proposed [11] initially by Roberts in a personal communication to Griggs. Interestingly, according to [15], essentially the same concept was also introduced by Harary in a private communication [14]. In fact, if we view labels as colors, then an $L(2,1)$-labelling is a *radio coloring* in the sense of [14, 15] and vice versa. In [14, 15], the minimum n for which there exists a radio coloring of Γ using colors from $\{1, 2, \ldots, n\}$ (not every color in $\{1, 2, \ldots, n\}$ needs to be used) is called the *radio coloring number* of Γ. Clearly, this number is exactly $\lambda(\Gamma) + 1$ for any graph Γ. In [14, 27] the minimum number of colors used in a radio coloring of Γ is called the *radio chromatic number* of Γ. From this definition it follows immediately that the radio chromatic number of Γ is exactly $\mu_{2,1}(\Gamma)$, and hence is equal to $\mu(\Gamma)$ by (1). Taking nonnegative integers as colors, a proper vertex coloring of the *square* Γ^2 of Γ is an $L(1,1)$-labelling of Γ and vice versa, where Γ^2 is defined to have vertex set $V(\Gamma)$ and edges joining distinct vertices of distance at most 2 in Γ. Thus, we have $\mu(\Gamma) = \chi(\Gamma^2)$, where χ is the chromatic number. Also, we notice that the invariant $\chi_{\bar{2}}(\Gamma)$ introduced in [28] is the same as $\mu(\Gamma)$.

In [11] Griggs and Yeh conjectured that $\lambda(\Gamma) \leq \Delta^2$ for any graph Γ with maximum degree $\Delta = \Delta(\Gamma) \geq 2$. In the same paper they proved that $\lambda(\Gamma) \leq \Delta^2 + 2\Delta$ for any graph Γ. This conjecture stimulated substantially the study of λ-number, and it was confirmed for quite a few classes of graphs, e.g., the class of graphs of diameter 2 considered in [11] and the class of chordal graphs [24]. For certain subclasses of chordal graphs the upper bound Δ^2 can be improved, as shown in [3]. For general graphs Γ, as far as we know, currently the best known bound is $\lambda(\Gamma) \leq \Delta^2 + \Delta - 1$ [20], which is an improvement of the bound $\lambda(\Gamma) \leq \Delta^2 + \Delta$ given in [3]. In the complexity aspect, Griggs and Yeh [11] proved that the $L(2,1)$-labelling problem is NP-complete for general graphs, and in contrast Chang and Kuo [3] gave a polynomial algorithm for trees.

The motivation of the present paper comes from the research [9, 29] on the λ-numbers of hypercubes and Hamming graphs. The *Cartesian product* $\Gamma_1 \Box \Gamma_2 \Box \cdots \Box \Gamma_d$

of $d \geq 2$ given graphs $\Gamma_1, \Gamma_2, \ldots, \Gamma_d$ is the graph with vertex set $V(\Gamma_1) \times V(\Gamma_2) \times \cdots \times V(\Gamma_d)$ such that $(\alpha_1, \alpha_2, \ldots, \alpha_d), (\beta_1, \beta_2, \ldots, \beta_d) \in V(\Gamma_1) \times V(\Gamma_2) \times \cdots \times V(\Gamma_d)$ are adjacent if and only if $\alpha_i \neq \beta_i$ for exactly one subscript i, and for such an i, α_i, β_i are adjacent in Γ_i. Let K_n denote the complete graph of order n. The Cartesian product

$$H_{n_1,n_2,\ldots,n_d} := K_{n_1} \Box K_{n_2} \Box \cdots \Box K_{n_d}$$

of complete graphs is called a *Hamming graph*, where $n_i \geq 2$ for each $i = 1, 2, \ldots, d$. As a convention, when we write $H_{n_1,n_2,\ldots,n_d}$ we assume without loss of generality that

$$n_1 \geq n_2 \geq \cdots \geq n_d \geq 2.$$

In the case where $n_1 = n_2 = \cdots = n_d = n$, we use $H(d, n)$ in place of $H_{n_1,n_2,\ldots,n_d}$. Thus,

$$H(d, n) := K_n \Box K_n \Box \cdots \Box K_n \ \ (d \text{ factors}).$$

In particular,

$$Q_d := H(d, 2)$$

is called the *d-cube* (hypercube). By using coding theory, Whittlesey, Georges, and Mauro [29, Theorem 3.7] proved that, if $2^{n-1} \leq d \leq 2^n - q$ for some q between 1 and $n+1$, then

$$\lambda(Q_d) \leq 2^n + 2^{n-q+1} - 2. \tag{2}$$

Recently, Georges, Mauro, and Stein [9] determined the λ-number of the Hamming graph $H(d, p^r)$ under the assumption that p is a prime, and either $d \leq p$ and $r \geq 2$, or $d < p$ and $r = 1$. They proved [9, Theorem 3.1] that, under these conditions,

$$\lambda(H(d, p^r)) = p^{2r} - 1. \tag{3}$$

In the same paper [9] they also determined the $\lambda_{j,k}$-number of H_{n_1,n_2}, and this work was extended to $H(3, n)$ in [8].

2. Main results.

Stimulated by [9, 29], our initial attempt was to improve the bound (2) and determine the λ-number of general Hamming graphs $H_{n_1,n_2,\ldots,n_d}$. This led us to a general approach to $L(j, k)$-labelling Cayley graphs on Abelian groups, which can be used to produce upper bounds for the $\lambda_{j,k}$-number and the μ-number of such graphs. In this section we will outline this approach and present the main results of the paper; see Theorems 2.2, 2.5, and 2.9 and their corollaries below. We will leave a detailed discussion on the approach to section 3. The approach seems to be powerful enough to derive the exact value of, or good upper bounds for, the $\lambda_{j,k}$-number and the μ-number of some Cayley graphs. In this paper we will apply it to Hamming graphs and a family of graphs containing all hypercubes. As we will see, (2) and (3) are special cases of our much more general results for such graphs.

Let G be a group and X a subset of G. If $1 \notin X$ and $x \in X$ implies $x^{-1} \in X$, where 1 is the identity element of G, then we call X a *Cayley set* of G. For such an X, the *Cayley graph* of G with respect to X, denoted by $\Gamma(G, X)$, is the graph with vertices the elements of G in which $x, y \in G$ are adjacent if and only if $xy^{-1} \in X$. Since X is inverse-closed, $\Gamma(G, X)$ is well defined as an undirected simple graph. To exclude the less interesting case where $\Gamma(G, X) = K_{|G|}$ is a complete graph, *we will* *assume without mentioning explicitly that* $X \neq G - \{1\}$. As usual, for a normal subgroup H of G, we use G/H to denote the quotient group of G by H, and $|G : H|$ the order of G/H. For any subsets X, Y of G, denote $XY := \{xy : x \in X, y \in Y\}$ and set $X^2 := XX$. As usual we use $\langle X \rangle$ to denote the subgroup of G generated by X. Call X a *generating set* of G if $\langle X \rangle = G$.

The key concept for our approach is the following definition of avoidability. Note that, for any Cayley set X of a group G, we have $1 = xx^{-1} \in X^2$ by the assumption that X is closed under taking inverse.

DEFINITION 2.1. *Let G be a finite Abelian group and X a Cayley set of G. A subgroup H of G is said to* avoid X *if $H \cap X = \emptyset$ and $H \cap X^2 = \{1\}$.*

Regarding this concept a few observations will be given in Remark 3.1. The following theorem shows that, once a subgroup H avoiding X is known, we can obtain upper bounds for the $\lambda_{j,k}$-number and the μ-number of $\Gamma(G, X)$.

THEOREM 2.2. *Let $j \geq k \geq 1$ be integers. Let G be a finite Abelian group and X a Cayley set of G. Then, for any subgroup H of G which avoids X, we have*

$$(4)\lambda_{j,k}(\Gamma(G,X)) \leq |G : H| \max\{k, \lceil j/2 \rceil\} + |G : \langle G - HX \rangle| \min\{j-k, \lfloor j/2 \rfloor\} - j$$

$$\mu(\Gamma(G,X)) \leq |G : H|. \tag{5}$$

A very important case occurs when $2k \geq j$. In this case we have $\max\{k, \lceil j/2 \rceil\} = k$, $\min\{j-k, \lfloor j/2 \rfloor\} = j-k$, and hence (4) becomes

$$\lambda_{j,k}(\Gamma(G,X)) \leq |G : H|k + |G : \langle G - HX \rangle|(j-k) - j. \tag{6}$$

In particular, for $L(2,1)$-labellings we have $2k = j = 2$ and hence Theorem 2.2 has the following consequence.

COROLLARY 2.3. *Let G be a finite Abelian group and X a Cayley set of G. Then, for any subgroup H of G which avoids X, we have*

$$\lambda(\Gamma(G,X)) \leq |G : H| + |G : \langle G - HX \rangle| - 2 \tag{7}$$

and

$$\mu(\Gamma(G,X)) \leq |G : H|.$$

An $L(j,k)$-labelling is called *no-hole* if the labels used by it consist of a set of consecutive integers. In the case where $G - HX$ is a generating set of G, we have $|G : \langle G - HX \rangle| = 1$, and Theorem 2.2 together with its proof implies the following result, which will be the main tool in our treatment of Hamming graphs.

COROLLARY 2.4. *Let $j \geq k \geq 1$ be integers. Let G be a finite Abelian group and X a Cayley set of G. Then, for any subgroup H of G which avoids X and is such that $G - HX$ generates G, we have*

$$\lambda_{j,k}(\Gamma(G,X)) \leq (|G : H| - 1) \max\{k, \lceil j/2 \rceil\}. \tag{8}$$

In particular,

$$\lambda(\Gamma(G,X)) \leq |G : H| - 1 \tag{9}$$

and $\Gamma(G,X)$ admits a no-hole $L(2,1)$-labelling using $|G : H|$ labels.

The class of Cayley graphs on Abelian groups is very large, and our results above apply to all such graphs universally. Because of this nature, it is unrealistic to expect that the bounds (4)–(9) are tight universally for all graphs in the class. However, as we will see later, for some Cayley graphs on Abelian groups they do produce sharp or near-sharp bounds for $\lambda_{j,k}$ and/or μ.

Let $\mathrm{Aut}(\Gamma)$ denote the automorphism group of a graph Γ. A subgroup G of $\mathrm{Aut}(\Gamma)$ is said to be *vertex-transitive* if, for any $\alpha, \beta \in V(\Gamma)$, there exists $g \in G$ such that g permutes α to β; such a group G is *regular* if there exists exactly one element g permuting α to β. The graph Γ is said to be *vertex-transitive* if $\mathrm{Aut}(\Gamma)$ is vertex-transitive. Using our general approach we obtain the following theorem for the family of connected graphs with automorphism group containing a vertex-transitive Abelian subgroup. Hypercubes are examples of such graphs (see the discussion after Corollary 2.6); for existence and construction of other graphs in this family, the reader is referred to [16, 17, 18]. For any integer $d \geq 1$, denote

$$n := 1 + \lfloor \log_2 d \rfloor$$

and

$$t := \min\{2^n - d - 1, n\}.$$

Note that both n and t are functions of d. From the definition of n it follows that $2^{n-1} \leq d < 2^n$, that is, n is the smallest integer such that $d < 2^n$. This choice of n makes the following upper bounds (10)–(16) as small as possible.

Theorem 2.5. *Let Γ be a connected graph whose automorphism group contains a vertex-transitive Abelian subgroup. Let d be the degree of vertices of Γ, and n, t be as above. Then, for any integers $j \geq k \geq 1$, we have*

$$\lambda_{j,k}(\Gamma) \leq 2^n \max\{k, \lceil j/2 \rceil\} + 2^{n-t} \min\{j-k, \lfloor j/2 \rfloor\} - j, \tag{10}$$

$$\mu(\Gamma) \leq 2^n. \tag{11}$$

As in (6), when $2k \geq j$, (10) becomes

$$\lambda_{j,k}(\Gamma) \leq 2^n k + 2^{n-t}(j-k) - j.$$

In particular, for $L(2,1)$-labellings, Theorem 2.5 implies the following corollary.

Corollary 2.6. *Let Γ and d be the same as in Theorem 2.5. Then*

$$\lambda(\Gamma) \leq 2^n + 2^{n-t} - 2 \tag{12}$$

and

$$\mu(\Gamma) \leq 2^n.$$

Note that Q_d is a Cayley graph on the elementary Abelian 2-group $\mathbb{Z}_2^d$ of order 2^d, namely $Q_d \cong \Gamma(\mathbb{Z}_2^d, X)$, where X is the set of elements of $\mathbb{Z}_2^d$ with exactly one nonzero coordinate. Thus, from [1, Lemma 16.3] it follows that Q_d admits $\mathbb{Z}_2^d$ as a vertex-transitive (regular, in fact) group of automorphisms. Since $\mathbb{Z}_2^d$ is Abelian, Theorem 2.5 and Corollary 2.6 imply the following two corollaries for Q_d.

Corollary 2.7. *Let d, j and k be integers with $d \geq 1$ and $j \geq k \geq 1$. Then*

$$\lambda_{j,k}(Q_d) \leq 2^n \max\{k, \lceil j/2 \rceil\} + 2^{n-t} \min\{j-k, \lfloor j/2 \rfloor\} - j \tag{13}$$

$$\mu(Q_d) \le 2^n. \tag{14}$$

Moreover, the proof of Theorem 2.5 gives rise to a systematic way of generating $L(j,k)$-labellings of Q_d which use 2^n labels and have span the right-hand side of (13); see the last paragraph of section 4. Again, when $2k \ge j$, (13) becomes

$$\lambda_{j,k}(Q_d) \le 2^n k + 2^{n-t}(j-k) - j.$$

In particular, for the λ-number of hypercubes, we have the following corollary.

Corollary 2.8. *For any integer $d \ge 1$, we have*

$$\lambda(Q_d) \le 2^n + 2^{n-t} - 2 \ \ ([29, \text{Theorem 3.7}]) \tag{15}$$

$$\mu(Q_d) \le 2^n \ \ ([28]). \tag{16}$$

The bounds (15) and (16) are equivalent to (2) and one of the main results of [28, line 12, pp. 185], respectively. To see this we distinguish the following two cases:

(i) $2^{n-1} \le d \le 2^n - n - 1$;

(ii) $2^n - n - 1 \le d \le 2^n - q$, for some q between 1 and n.

In case (i), $t = n$ and we may choose $q = n+1$ in (2); hence $t = q-1$ and (15) and (2) are identical. Also, in this case the upper bounds in (10) and (13) are $(2^n - 1)\max\{k, \lceil j/2 \rceil\}$, and that in (12) and (15) are $2^n - 1$. In case (ii), we have $q - 1 \le 2^n - d - 1 \le n$; hence $t = 2^n - d - 1$ and (15) and (2) are the same if we choose $q = 2^n - d$.

The bound (14) is tight when $d = 2^n - 1$. In fact, for any $d \ge 1$, since the d neighbors of the 0-labelled vertex of Q_d are distance two apart, they must be assigned distinct labels no less than j under any $L(j,k)$-labelling. Thus, $\mu(Q_d) \ge d+1$. In the case where $d = 2^n - 1$, we have $\mu(Q_d) \le d+1$ by (14) and hence $\mu(Q_d) = d+1$, that is, (14) is sharp. Note that (15) implies $\lambda(Q_d) \le 2d$, as noticed in [29, Theorem 3.8].

For Hamming graphs we obtain the following results by using Theorem 2.2.

Theorem 2.9. *Let n_1, n_2, d be integers such that $n_1 > d \ge 2$, n_2 divides n_1, and each prime factor of n_1 is no less than d. Then, for any integers $j \ge k \ge 1$, and for any positive integers $n_3, \dots, n_d$ which are less than or equal to n_2, we have*

$$\lambda_{j,k}(H_{n_1,n_2,\dots,n_d}) \le (n_1 n_2 - 1)\max\{k, \lceil j/2 \rceil\} \tag{17}$$

$$\mu(H_{n_1,n_2,\dots,n_d}) = n_1 n_2 \tag{18}$$

and we can give explicitly an $L(j,k)$-labelling of $H_{n_1,n_2,\dots,n_d}$ which has span $(n_1 n_2 - 1)\max\{k, \lceil j/2 \rceil\}$ and is optimal for μ. Furthermore, if in addition $2k \ge j$, then

$$\lambda_{j,k}(H_{n_1,n_2,\dots,n_d}) = (n_1 n_2 - 1)k \tag{19}$$

and this $L(j,k)$-labelling is optimal for $\lambda_{j,k}$ and μ simultaneously.

Note that, in the case where $2k \ge j$, Theorem 2.9 gives the exact values of both $\lambda_{j,k}$ and μ for $H_{n_1,n_2,\dots,n_d}$ above. It shows that the trivial lower bounds $\lambda_{j,k}(H_{n_1,n_2,\dots,n_d}) \ge (n_1 n_2 - 1)k$ and $\mu(H_{n_1,n_2,\dots,n_d}) \ge n_1 n_2$ (see Lemma 5.1) are both obtained. Another interesting feature is that both $\lambda_{j,k}$ and μ are irrelevant to j in this case: they rely on k, n_1, and n_2 only. In particular, for the $L(2,1)$ case we have the following corollary.

Corollary 2.10. *Let $n_1, n_2, n_3, \ldots, n_d$ and $d \geq 2$ be as in Theorem* 2.9. *Then*

$$\lambda(H_{n_1,n_2,\ldots,n_d}) = n_1 n_2 - 1 \tag{20}$$

$$\mu(H_{n_1,n_2,\ldots,n_d}) = n_1 n_2. \tag{21}$$

Moreover, we can give explicitly a no-hole $L(2,1)$-labelling of $H_{n_1,n_2,\ldots,n_d}$ which is optimal for λ and μ simultaneously.

For special Hamming graphs $H(d,n)$ (which is the graph K_n^d in [9]), Theorem 2.9 implies the following result.

Corollary 2.11. *Let $n = p_1^{r_1} p_2^{r_2} \cdots p_t^{r_t}$, where p_i is a prime and $r_i \geq 1$, for each $i = 1, 2, \ldots, t$. Let d be an integer such that $2 \leq d \leq p_i$ for each i and $\sum_{i=1}^{t}(p_i - d + r_i) \geq 2$. Then, for any integers $j \geq k \geq 1$, we have*

$$\lambda_{j,k}(H(d,n)) \leq (n^2 - 1)\max\{k, \lceil j/2 \rceil\}$$

and

$$\mu(H(d,n)) = n^2.$$

Moreover, if in addition $2k \geq j$, then

$$\lambda_{j,k}(H(d,n)) = (n^2 - 1)k. \tag{22}$$

The condition $\sum_{i=1}^{t}(p_i - d + r_i) \geq 2$ ensures that $n > d$, as required by Theorem 2.9. It is equivalent to either $t \geq 2$, or $t = 1$ and $p_1 - d + r_1 \geq 2$. In the latter case, $n = p^r$ is a prime power and (22) becomes $\lambda_{j,k}(H(d,p^r)) = (p^{2r} - 1)k$. For the $L(2,1)$ case, this gives $\lambda(H(d,p^r)) = p^{2r} - 1$, which is exactly (3). Also, we can get (3) from (20) directly. Thus, Corollaries 2.10–2.11 (and hence Theorem 2.9) generalize (3) to a wide extent.

Theorems 2.2, 2.5, and 2.9 will be proved in sections 3, 4, and 5, respectively. Remarks on the results above will be given in these sections as well. Concluding remarks and open questions arising from Theorem 2.9 will be offered in the last section.

3. Proof of Theorem 2.2.

3. Proof of Theorem 2.2. The terminology and notation for groups used in the paper are standard; see, for example, [25]. We will reserve the upper case English letters G, H for groups and the upper case Greek letters Γ, Σ for graphs. We will use certain lower case English letters such as g, h, u, v, w, x, y, z to denote elements of groups, but we reserve $d, i, j, k, \ell, m, n, r, s, t$ for integers. For two sets X and Y, $X - Y$ denotes the set $\{x \in X : x \notin Y\}$. For any graph Γ and a partition $\mathcal{P}$ of $V(\Gamma)$, the *quotient graph* $\Gamma_{\mathcal{P}}$ of Γ with respect to $\mathcal{P}$ is defined to have vertex set $\mathcal{P}$ in which two parts of $\mathcal{P}$ are adjacent if and only if there exists at least one edge of Γ joining a vertex in the first part to a vertex in the second part. In the case where each part of $\mathcal{P}$ is an independent set of Γ with ℓ vertices, for some integer $\ell \geq 1$, and the subgraph induced by two adjacent parts is a perfect matching of ℓ edges, the graph Γ is called an *ℓ-fold cover* of the quotient $\Gamma_{\mathcal{P}}$.

Let G be a finite group. For an element x of G, we will use $o(x)$ to denote the *order* of x in G, that is, the smallest positive integer n such that $x^n = 1$. The element x is called an *involution* if $o(x) = 2$. For a Cayley set X of G, from the definition of $\Gamma(G,X)$ it follows that $x, y \in G$ are connected by a path of $\Gamma(G,X)$ if and only if $xy^{-1} \in \langle X \rangle$; in particular $\Gamma(G,X)$ is a connected graph if and only if $\langle X \rangle = G$. Moreover, $\Gamma(G,X)$ is vertex-transitive and G is isomorphic to a regular subgroup of the automorphism group of $\Gamma(G,X)$ (see, e.g., [1, Theorem 16.4]). In particular, all vertices of $\Gamma(G,X)$ have the same degree, which is equal to $|X|$. For a normal subgroup H of G, the quotient group G/H gives rise to a natural partition of G with parts the cosets Hg of H in G. We will use the same notation G/H for this partition. Denote $X/H := \{Hx : x \in X\}$. Then

$$X/H = \{Hz \in G/H : Hz \cap X \neq \emptyset\}.$$

It should be noticed that X/H is not necessarily a subgroup of the quotient group G/H, and that $Hx \in X/H$ does not imply $x \in X$.

The idea behind our approach is rather natural: for a Cayley graph $\Gamma(G,X)$ on an Abelian group G, if we can find a subgroup H of G which "avoids" the Cayley set X, then we can label the elements in the same coset of H in G by the same label. In this way we get an $L(j,k)$-labelling of $\Gamma(G,X)$ and thus upper bounds for $\lambda_{j,k}(\Gamma(G,X))$ and $\mu(\Gamma(G,X))$. A very special case of this method for $L(2,1)$-labelling Hamming graphs $H(d,p^r)$ was used implicitly in the proof of [9, Theorem 3.1]. The approach proposed in the present paper is much more general and powerful. Before proceeding to the proof of Theorem 2.2, let us record the following observations about the concept of avoidability.

Remark 3.1. (a) The trivial subgroup $\{1\}$ avoids every Cayley set of G.

(b) The condition $H \cap X^2 = \{1\}$ implies that either $H \cap X = \emptyset$ or $H \cap X = \{x\}$ for an involution x of G. In fact, if $H \cap X \neq \emptyset$, then $xy = 1$ for any $x, y \in H \cap X$ (not necessarily distinct) since $xy \in H \cap X^2 = \{1\}$. That is, any two elements of $H \cap X$ are inverse of each other. From this it follows that $H \cap X = \{x\}$ for an involution x of G.

(c) Thus, if G contains no involutions, then H avoids X if and only if $H \cap X^2 = \{1\}$. This is the case in particular when, say, the order of G is odd.

To prove Theorem 2.2 we need some combinatorial properties of the Cayley graph $\Gamma(G,X)$ and its quotient graph $(\Gamma(G,X))_{G/H}$ with respect to the partition G/H, where $H \leq G$ avoids X. Define

$$\mathcal{G}_{H,X} := \{Hz \in G/H : Hz \cap X = \emptyset\}. \tag{23}$$

Since H avoids X, we have $H \cap X = \emptyset$, and hence $H \in \mathcal{G}_{H,X}$ and $H \subseteq G - HX$. (In fact, if $H \not\subseteq G - HX$, then $h_1 = h_2 x$ for some $h_1, h_2 \in H$, $x \in X$, and hence $h_2^{-1} h_1 = x \in H \cap X = \emptyset$, a contradiction.) Thus, $\mathcal{G}_{H,X} \neq \emptyset$ and $H \leq \langle G - HX \rangle$. Also, $Hz \in \mathcal{G}_{H,X}$ if and only if $x \notin Hz$ for all $x \in X$, which is true if and only if $Hz \neq Hx$ for all $x \in X$. Therefore, we have

$$\mathcal{G}_{H,X} = G/H - X/H = (G - HX)/H \tag{24}$$

and hence

$$\langle \mathcal{G}_{H,X} \rangle = \langle G - HX \rangle / H. \tag{25}$$

Lemma 3.2. *Let G be a finite Abelian group and X a Cayley set of G. Let H be a subgroup of G which avoids X. Then the following* (a)–(d) *hold.*

(a) *The mapping ψ defined by $x \mapsto Hx$, for $x \in X$, is a bijection from X to X/H.*

(b) *Any two vertices in the same coset of H in G are at least distance three apart in $\Gamma(G, X)$; in particular each coset of H is an independent set of $\Gamma(G, X)$.*

(c) *Both X/H and $\mathcal{G}_{H,X} - \{H\}$ are Cayley sets of G/H; moreover, the corresponding Cayley graphs $\Gamma(G/H, X/H)$, $\Gamma(G/H, \mathcal{G}_{H,X} - \{H\})$ are complementary graphs with degrees $|X|$, $|G : H| - |X| - 1$, respectively.*

(d) $\Gamma(G/H, X/H) \cong (\Gamma(G, X))_{G/H}$, *and* $\Gamma(G, X)$ *is an* $|H|$*-fold cover of* $\Gamma(G/H, X/H)$.

Proof. (a) Clearly, ψ is surjective. If $Hx = Hy$ for distinct $x, y \in X$, then $1 \neq xy^{-1} \in H \cap X^2$, which contradicts the avoidability of H from X. Thus, ψ is also injective and hence is a bijection from X to X/H.

(b) For distinct $x, y \in G$ in the same coset of H, we have $xy^{-1} \in H - \{1\}$. Thus, since H avoids X, we have $xy^{-1} \notin X \cup X^2$. By the definition of $\Gamma(G, X)$, it is easy to see that the distance $d(x, y)$ in $\Gamma(G, X)$ between x and y is equal to the minimum number of elements of X whose product is xy^{-1}. Therefore, $xy^{-1} \notin X \cup X^2$ implies $d(x, y) \geq 3$, as required.

(c) Since X is a Cayley set of G, it is closed under taking inverse. This together with the fact that $(Hx)^{-1} = Hx^{-1}$ implies that X/H is closed under taking inverse as well. Also, since $H \cap X = \emptyset$, the identity H of G/H is not in X/H. Thus, X/H is a Cayley set of G/H. Since $\mathcal{G}_{H,X} - \{H\} = G/H - X/H - \{H\}$ by (24), this implies that $\mathcal{G}_{H,X} - \{H\}$ is a Cayley set of G/H as well. Note that X/H and $\mathcal{G}_{H,X} - \{H\}$ constitute a partition of G/H. Therefore, they give rise to complementary Cayley graphs of G/H. From (a) we have $|X/H| = |X|$, and hence $\Gamma(G/H, X/H)$ has degree $|X|$. Consequently, $\Gamma(G/H, \mathcal{G}_{H,X} - \{H\})$ has degree $|G : H| - |X| - 1$.

(d) We have $Hx, Hy \in G/H$ are adjacent in $\Gamma(G/H, X/H) \Leftrightarrow Hx(Hy)^{-1} \in X/H$ $\Leftrightarrow H(xy^{-1}) = Hz$ for some $z \in X \Leftrightarrow xy^{-1} = hz$ for some $z \in X$ and $h \in H \Leftrightarrow$ $x(hy)^{-1} = z$ for some $z \in X$ and $h \in H \Leftrightarrow x \in Hx$ and $hy \in Hy$ are adjacent in $\Gamma(G, X)$ for some $h \in H \Leftrightarrow gx \in Hx$ and $ghy \in Hy$ are adjacent in $\Gamma(G, X)$ for some $h \in H$ and any $g \in H \Leftrightarrow Hx, Hy$ are adjacent in the quotient graph $(\Gamma(G, X))_{G/H}$. (Here we used the assumption that G is Abelian.) Hence we have $\Gamma(G/H, X/H) \cong (\Gamma(G, X))_{G/H}$. Moreover, from the arguments above we see that, for adjacent cosets Hx and Hy, each element of Hx is adjacent to at least one element of Hy in $\Gamma(G, X)$. However, $\Gamma(G, X)$ and $\Gamma(G/H, X/H)$ have the same degree $|X|$. So the subgraph of $\Gamma(G, X)$ induced by $Hx \cup Hy$ is forced to be a perfect matching between Hx and Hy. Therefore, $\Gamma(G, X)$ is an $|H|$-fold cover of $\Gamma(G/H, X/H)$. □

In the case where in addition $\langle X \rangle = G$, one can check that $\Gamma(G/H, X/H)$ is the underlying undirected graph of the Schreier coset graph for (G, H, X), and in this case this Schreier coset graph has no loop or multiple arc. (For any group G with generating set X, and any subgraph H of G, the *Schreier coset graph* [12] for (G, H, X) is the directed graph with vertex set $G/H = \{Hz : z \in G\}$ and arcs (Hz, Hzx) for all Hz and $x \in X$, where loops and multiple arcs are allowed.)

A cycle (path, respectively) in a graph visiting all vertices is called a Hamiltonian cycle (Hamiltonian path, respectively). A graph is *Hamiltonian* if it contains a Hamiltonian cycle. The following result is well known; see, e.g., [21, Corollary 3.2].

LEMMA 3.3. *Every connected Cayley graph on a finite Abelian group of order at least three is Hamiltonian.*

An immediate consequence of this result is that every connected Cayley graph on any finite Abelian group contains a Hamiltonian path. This will be used in the following proof of Theorem 2.2.

Proof of Theorem 2.2. Let G be a finite Abelian group and X a Cayley set of

G. Let H be a subgroup of G which avoids X. For notational simplicity, we denote $\mathcal{G} = \langle \mathcal{G}_{H,X} \rangle$ and $\hat{x} = Hx$ for $x \in G$. Denote $r = |G : H|$ and $s = |G : \langle G - HX \rangle|$. Then $s = |(G/H) : \mathcal{G}| = r/|\mathcal{G}|$ by (25).

Let us first treat the degenerate case where $\mathcal{G}_{H,X} = \{H\}$. In this case we have $s = r$ and $X/H = G/H - \{H\}$, and hence $\Gamma(G/H, X/H)$ is a complete graph. Order linearly the cosets in G/H in an arbitrary way. Then assign label $(i-1)j$ to every element of the ith member of G/H, for $i = 1, 2, \ldots, r$. Using Lemma 3.2(b) and noting $j \geq k$, one can check that this labelling is an $L(j,k)$-labelling of $\Gamma(G,X)$. Clearly, it uses r labels and has span $(r-1)j$. Thus, we have $\lambda_{j,k}(\Gamma(G,X)) \leq (r-1)j$ and $\mu(\Gamma(G,X)) \leq r$. But, since $s = r$ and $\max\{k, \lceil j/2 \rceil\} + \min\{j-k, \lfloor j/2 \rfloor\} = j$, the right-hand side of (4) is exactly $(r-1)j$. Therefore, we have proved (4) and (5) in the case where $\mathcal{G}_{H,X} = \{H\}$.

In the following we deal with the general case where $\mathcal{G}_{H,X} - \{H\} \neq \emptyset$. Let

$$\mathcal{G}\hat{x}_1, \mathcal{G}\hat{x}_2, \ldots, \mathcal{G}\hat{x}_s$$

be representatives of distinct cosets of $\mathcal{G}$ in G/H, where we set $\mathcal{G}\hat{x}_1 = \mathcal{G}$. Then of course they consist of a partition of G/H. Recall from Lemma 3.2(c) that $\mathcal{G}_{H,X} - \{H\}$ is a Cayley set of G/H. By the definition of the Cayley graph $\Gamma(G/H, \mathcal{G}_{H,X} - \{H\})$, two cosets $\hat{x}, \hat{y}$ of H are connected by a path of $\Gamma(G/H, \mathcal{G}_{H,X} - \{H\})$ if and only if $\hat{x}(\hat{y})^{-1} = \widehat{xy^{-1}} \in \langle \mathcal{G}_{H,X} - \{H\} \rangle = \mathcal{G}$, which in turn is true if and only if $\hat{x}, \hat{y}$ are in the same coset $\mathcal{G}\hat{x}_i$ of $\mathcal{G}$ in G/H, for some i. Thus, for each $i = 1, 2, \ldots, s$, $\mathcal{G}\hat{x}_i$ induces a connected component of $\Gamma(G/H, \mathcal{G}_{H,X} - \{H\})$. In what follows we will denote this component by $\widehat{\Gamma}_i$. These components $\widehat{\Gamma}_i$, $i = 1, 2, \ldots, s$, are isomorphic to each other since as a Cayley graph $\Gamma(G/H, \mathcal{G}_{H,X} - \{H\})$ is vertex-transitive. Since $\mathcal{G}_{H,X} - \{H\}$ generates $\mathcal{G}$ and is a Cayley set of G/H (Lemma 3.2(c)), it is also a Cayley set of $\mathcal{G}$. Hence $\mathcal{G}_{H,X} - \{H\}$ gives rise to a connected Cayley graph $\Gamma(\mathcal{G}, \mathcal{G}_{H,X} - \{H\})$, which is exactly the connected component $\widehat{\Gamma}_1$ of $\Gamma(G/H, \mathcal{G}_{H,X} - \{H\})$ induced by $\mathcal{G}$. By Lemma 3.3, $\widehat{\Gamma}_1$ contains a Hamiltonian path, and hence so does each $\widehat{\Gamma}_i$ as $\widehat{\Gamma}_i \cong \widehat{\Gamma}_1$. Let

$$\hat{x}_{i,1}, \hat{x}_{i,2}, \ldots, \hat{x}_{i,t}$$

be a Hamiltonian path of $\widehat{\Gamma}_i$, where $t = |\mathcal{G}| = r/s$. Then any two consecutive members in this sequence are adjacent in $\Gamma(G/H, \mathcal{G}_{H,X} - \{H\})$, and hence are not adjacent in $\Gamma(G/H, X/H)$ by Lemma 3.2(c). Hence, for each $i = 1, 2, \ldots, s$, by Lemma 3.2(d) there is no edge of $\Gamma(G,X)$ joining any element of $\hat{x}_{i,\ell}$ and any element of $\hat{x}_{i,\ell+1}$, for $\ell = 1, 2, \ldots, t-1$. By Lemma 3.2(b) the elements of $\hat{x}_{i,\ell}$ are distance at least three apart in $\Gamma(G,X)$, for each i and $\ell = 1, 2, \ldots, t$.

Now we define f to be the labelling such that all the elements of $\hat{x}_{i,\ell}$ are labelled by

$$(i-1)\left((t-1)\max\{k, \lceil j/2 \rceil\} + j\right) + (\ell - 1)\max\{k, \lceil j/2 \rceil\}$$

for $i = 1, 2, \ldots, s$ and $\ell = 1, 2, \ldots, t$. Then, for any $\hat{x}_{i,\ell}$ and $\hat{x}_{i',\ell'}$ with $i \neq i'$, the labels of the elements of $\hat{x}_{i,\ell}$ and $\hat{x}_{i',\ell'}$ differ by at least j. For $\hat{x}_{i,\ell}$ and $\hat{x}_{i,\ell'}$ with the same first subscript, if an element of $\hat{x}_{i,\ell}$ is adjacent to an element of $\hat{x}_{i,\ell'}$ in $\Gamma(G,X)$, then $|\ell - \ell'| \geq 2$ by the discussion in the previous paragraph, and hence the labels of these two elements differ by at least $2\max\{k, \lceil j/2 \rceil\}$, which is obviously no less than j. Also, if an element of $\hat{x}_{i,\ell}$ is distance two apart from an element of $\hat{x}_{i,\ell'}$ in $\Gamma(G,X)$, then $\ell \neq \ell'$ by Lemma 3.2(b) and hence the labels of these two elements differ by at least $\max\{k, \lceil j/2 \rceil\}$, which is no less than k. Therefore, f is an $L(j,k)$-labelling of $\Gamma(G,X)$. Noting that $r = st$, this labelling uses r distinct labels and has span

$$\begin{aligned}\mathrm{sp}(\Gamma(G,X);f) &= (s-1)((t-1)\max\{k,\lceil j/2\rceil\}+j)+(t-1)\max\{k,\lceil j/2\rceil\}\\ &= r\max\{k,\lceil j/2\rceil\}+s(j-\max\{k,\lceil j/2\rceil\})-j\\ &= r\max\{k,\lceil j/2\rceil\}+s\min\{j-k,\lfloor j/2\rfloor\}-j.\end{aligned}$$

Therefore, the upper bounds (4) and (5) follow and the proof is complete. □

Proof of Corollary 2.4. We use the notation in the proof of Theorem 2.2. Since H avoids X and $G - HX$ is a generating set of G, we have $\mathcal{G} = \langle \mathcal{G}_{H,X} \rangle = G/H$ by (25). Hence $s = 1$, $t = r = |G : H|$, and $\Gamma(G/H, \mathcal{G}_{H,X} - \{H\})$ is connected. Thus, by Lemma 3.3, $\Gamma(G/H, \mathcal{G}_{H,X} - \{H\})$ contains a Hamiltonian path $Hx_{1,1}, Hx_{1,2}, \ldots, Hx_{1,r}$. From the proof of Theorem 2.2, the labelling f which assigns $(\ell - 1)\max\{k, \lceil j/2 \rceil\}$ to the elements of $Hx_{1,\ell}$ $(\ell = 1, 2, \ldots, r)$ is an $L(j,k)$-labelling of $\Gamma(G,X)$. Since this labelling has span $(r-1)\max\{k, \lceil j/2 \rceil\}$, we obtain (8) immediately.

For the $L(2,1)$ case, we have $2k = j = 2$ and hence (9) follows from (8). Also, in this case the labelling f above uses labels $0, 1, 2, \ldots, r-1$, and hence is a no-hole $L(2,1)$-labelling. This completes the proof. □

We conclude this section by giving the following remarks.

Remark 3.4. (a) The proof of Theorem 2.2 gives an explicit $L(j,k)$-labelling of $\Gamma(G,X)$ provided that a Hamiltonian cycle of $\Gamma(\mathcal{G}, \mathcal{G}_{H,X} - \{H\})$ is known, where $\mathcal{G} = \langle \mathcal{G}_{H,X} \rangle$ as above.

(b) A Cayley set X may be avoided by several subgroups H of G. To get a better upper bound for $\lambda_{j,k}(\Gamma(G,X))$, we will be interested in those H such that the right-hand side of (4) is as small as possible.

In the case where $G - HX$ is a generating set of G, we have by (8)

$$\lambda_{j,k}(\Gamma(G,X)) \leq (|G:H|-1)\max\{k,\lceil j/2\rceil\} \leq (|G|-1)\max\{k,\lceil j/2\rceil\}.$$

Note that the second equality occurs precisely when $H = \{1\}$. On the other hand, if $H = \{1\}$, then $G - HX$ is a generating set of $G \Leftrightarrow G - X$ is a generating set of $G \Leftrightarrow$ the complement graph of $\Gamma(G,X)$ is connected $\Leftrightarrow$ the complement graph of $\Gamma(G,X)$ has a Hamiltonian path $\Leftrightarrow$ the elements of G can be ordered as $x_1, x_2, \ldots, x_{|G|}$ such that any two consecutive elements in this sequence are nonadjacent in $\Gamma(G,X)$. In this simplest case, (9) gives the bound $\lambda(\Gamma(G,X)) \leq |G| - 1$, which is the same as the one obtained by using [10, Theorem 1.1(a)]. The reader can easily find examples which show that even in this somewhat "worst" case the bound $|G| - 1$ can be the actual value of the λ-number of $\Gamma(G,X)$.

(c) The bound (7) can be improved as

$$\lambda(\Gamma(G,X)) \leq |G : \langle G - HX \rangle|(\lambda_0 + 2) - 2, \tag{26}$$

where $\lambda_0 = \lambda(\Gamma(\mathcal{G}, \mathcal{G} - \mathcal{G}_{H,X}))$. In fact, in the proof of Theorem 2.2 for the $L(2,1)$ case, we assigned t $(= |\mathcal{G}| = |\langle G - HX \rangle : H|)$ distinct labels to the vertices of $\widehat{\Gamma}_i$. But $\lambda_0 + 1$ labels will be enough, and so replacing t by $\lambda_0 + 1$ in the proof of Theorem 2.2 will give the proof of (26). Note that $\lambda_0 + 1 \leq t$ since the complementary graph $\Gamma(\mathcal{G}, \mathcal{G}_{H,X} - \{H\})$ of $\Gamma(\mathcal{G}, \mathcal{G} - \mathcal{G}_{H,X})$ contains a Hamiltonian path. Hence (26) does imply (7), and it is better than (7) in the case where $\lambda_0 + 1$ is strictly less than

t. The inequality (26) establishes a connection between the λ-numbers of $\Gamma(G, X)$ and $\Gamma(\mathcal{G}, \mathcal{G} - \mathcal{G}_{H,X})$, the latter being an induced subgraph of the quotient graph $\Gamma(G/H, X/H)$ of $\Gamma(G, X)$.

(d) From (25) one can see that (4) and (7) can be rewritten as

$$\lambda_{j,k}(\Gamma(G,X)) \leq |G:H| \left(\max\{k, \lceil j/2 \rceil\} + \frac{\min\{j-k, \lfloor j/2 \rfloor\}}{|\langle \mathcal{G}_{H,X} \rangle|} \right) - j \tag{27}$$

$$\lambda(\Gamma(G,X)) \leq |G:H| \left(1 + \frac{1}{|\langle \mathcal{G}_{H,X} \rangle|} \right) - 2, \tag{28}$$

respectively. As in (6), if $2k \geq j$, then $\max\{k, \lceil j/2 \rceil\}$ in (27) can be replaced by k.

(e) From (24) it follows that $\mathcal{G}_{H,X} = \{H\}$ occurs if and only if $\{H, HX\}$ is a partition of G. (Note that $H \cap X = \emptyset$ implies $H \cap HX = \emptyset$.) In this extreme case we have $\langle \mathcal{G}_{H,X} \rangle = \{H\}$ and hence (27) becomes

$$\lambda_{j,k}(\Gamma(G,X)) \leq (|G:H| - 1)j.$$

4. Proof of Theorem 2.5. To prove Theorem 2.5 we need the following well known result.

LEMMA 4.1 (see [1, Proposition 16.5]). *Let Γ be a graph whose automorphism group contains a vertex-transitive Abelian subgroup G. Then G is regular on $V(\Gamma)$, and G is an elementary Abelian 2-group.*

(Note that in [1] this proposition is stated for the full automorphism group $\mathrm{Aut}(\Gamma)$ of Γ. However, it is valid for a transitive Abelian subgroup of $\mathrm{Aut}(\Gamma)$ as well, and the proof is the same.)

In the following we will use $V(d, 2)$ to denote the d-dimensional linear space of row vectors over the field $\mathrm{GF}(2) = \{0, 1\}$ of characteristic 2, and $V^+(d, 2)$ to denote the additive group of $V(d, 2)$. For this group the operation is addition of row vectors, and hence we will use $H + x$ in place of Hx. Denote by $\mathbf{0}_d$ the zero vector of $V(d, 2)$. Then it is the identity element of $V^+(d, 2)$. It is well known that $V^+(d, 2)$ is isomorphic to the elementary Abelian 2-group $\mathbb{Z}_2^d$.

As we will soon see, any connected graph Γ with $\mathrm{Aut}(\Gamma)$ containing a vertex-transitive Abelian subgroup G is isomorphic to a Cayley graph on G. To prove Theorem 2.5 by using Theorem 2.2, we need to identify a subgroup of G such that it avoids the relevant Cayley set and produces the upper bounds (10) and (11). This is equivalent to identifying a subspace of $V(d, 2)$ with certain properties, and hence is a matrix problem essentially. The existence of such a subspace is guaranteed by the following lemma.

LEMMA 4.2. *Let d, ℓ, n be positive integers such that $n \leq \ell \leq d$ and $2^{n-1} \leq d < 2^n$. Let $t := \min\{2^n - d - 1, n\}$. Then for any d nonzero, pairwise distinct vectors $\mathbf{x}_1, \ldots, \mathbf{x}_d$ of $V(\ell, 2)$ which generate $V(\ell, 2)$, there exists an $\ell \times n$ matrix M over* $\mathrm{GF}(2)$ *such that*

(a) *M has rank n;*

(b) *$\mathbf{x}_1 M, \ldots, \mathbf{x}_d M$ are nonzero and pairwise distinct; and*

(c) *$V(n, 2) - \{\mathbf{x}_1 M, \ldots, \mathbf{x}_d M\}$ contains t independent vectors.*

Proof. Since $t \leq n$, we can choose t independent vectors $\mathbf{d}_1, \ldots, \mathbf{d}_t$ of $V(n, 2)$. Since $V(n, 2)$ has $2^n - 1$ nonzero vectors and $t + d \leq 2^n - 1$ by the definition of t, we can choose d distinct nonzero vectors, say $\mathbf{c}_1, \ldots, \mathbf{c}_d$, from $V(n, 2) - \{\mathbf{d}_1, \ldots, \mathbf{d}_t\}$. Moreover, we may require that the $d \times n$ matrix C with the ith row $\mathbf{c}_i$ has rank n, so that its columns are independent. For example, if $1 \leq t < n$, then we can set $\mathbf{d}_i$, for $1 \leq i \leq t$, to be the vector with the jth entry 0 if $j < i$ and 1 if $j \geq i$; if $t = n$, then we can set $\mathbf{d}_n$ to be $(1, 0, \ldots, 0, 1)$ and define other $\mathbf{d}_i$'s in the same way. (In the case where $t = 0$ we leave $\mathbf{d}_t$ undefined.) Set $\mathbf{c}_1 = (1, 0, \ldots, 0), \ldots, \mathbf{c}_n = (0, 0, \ldots, 1)$ to be the standard basis of $V(n, 2)$, and choose distinct nonzero vectors $\mathbf{c}_{n+1}, \ldots, \mathbf{c}_d$ from $V(n, 2) - \{\mathbf{c}_1, \ldots, \mathbf{c}_n, \mathbf{d}_1, \ldots, \mathbf{d}_t\}$. Then $\mathbf{c}_1, \ldots, \mathbf{c}_n, \mathbf{c}_{n+1}, \ldots, \mathbf{c}_d, \mathbf{d}_1, \ldots, \mathbf{d}_t$ satisfy all the conditions above. Moreover, the matrix C has the form

$$C = \begin{pmatrix} I_n \\ J \end{pmatrix},$$

where I_n is the identity matrix of order n over GF(2) and J is the $(d - n) \times n$ matrix of rows $\mathbf{c}_{n+1}, \ldots, \mathbf{c}_d$. Since $\ell \leq d$ and the columns of C are independent vectors of dimension d, we can add $\ell - n$ column vectors of dimension d to C to form a $d \times \ell$ matrix Y of rank ℓ. Thus, the columns of Y are independent, and the rows $\mathbf{y}_1, \ldots, \mathbf{y}_d$ of Y are extensions of $\mathbf{c}_1, \ldots, \mathbf{c}_d$, respectively, that is,

$$\mathbf{y}_i = (\mathbf{c}_i \mid \overbrace{*, \ldots, *}^{\ell - n})$$

for each i. Set

$$B = \begin{pmatrix} I_n \\ 0 \end{pmatrix},$$

where 0 is the $(\ell - n) \times n$ matrix with all entries zero. Then B is an $\ell \times n$ matrix of rank n, and it satisfies $YB = C$. Let A be the $d \times \ell$ matrix with the ith row $\mathbf{x}_i$, for $1 \leq i \leq d$. Then A has rank ℓ by our assumption. Since Y has also rank ℓ, from linear algebra there exists a nonsingular $\ell \times \ell$ matrix N over GF(2) such that $Y = AN$. Now we set $M = NB$. Then the nonsingularity of N ensures that M has the same rank as B, that is, M has rank n. Also, we have $AM = A(NB) = YB = C$, which implies $\mathbf{x}_i M = \mathbf{c}_i$ for each i. Thus, $\mathbf{x}_1 M, \ldots, \mathbf{x}_d M$ are nonzero and pairwise distinct. Moreover, $\mathbf{d}_1, \ldots, \mathbf{d}_t$ are t independent vectors in $V(n, 2) - \{\mathbf{x}_1 M, \ldots, \mathbf{x}_d M\}$. $\square$

Proof of Theorem 2.5. Let Γ be a connected graph such that $\mathrm{Aut}(\Gamma)$ contains a vertex-transitive Abelian subgroup G. By Lemma 4.1, G is regular on $V(\Gamma)$, and G is an elementary Abelian 2-group. Hence $|G| = 2^\ell$ and $G \cong \mathbb{Z}_2^\ell$ for a positive integer ℓ (see, e.g., [25, 7.40]). In the following we will identify G with the group $V^+(\ell, 2)$. Since G is regular on $V(\Gamma)$, by [1, Lemma 16.3] Γ is isomorphic to a Cayley graph of G, namely $\Gamma \cong \Gamma(G, X)$ for a Cayley set

$$X := \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$$

of G, where $d := |X|$ is the degree of vertices of Γ and each $\mathbf{x}_i \in V(\ell, 2)$. Moreover, X must be a generating set of G as Γ is connected. Hence $\ell \leq d$. Also, we have $d < 2^\ell$ as X is a proper subset of G. Let $n := 1 + \lfloor \log_2 d \rfloor$ and $t := \min\{2^n - d - 1, n\}$. Then $2^{n-1} \leq d < 2^n$ and hence $2^{n-1} \leq d < 2^\ell$, which implies $n \leq \ell$. From Lemma 4.2 there exists an $\ell \times n$ matrix M over GF(2) with properties (a)–(c) in that lemma. Since M has rank n by property (a) there, its null space

$$U := \{\mathbf{x} \in V(\ell, 2) : \mathbf{x}M = \mathbf{0}_n\}$$

is an $(\ell - n)$-dimensional subspace of $V(\ell, 2)$. Let $H := U^+$ be the additive group of U. Then $|G : H| = 2^n$. From the definition (23) of $\mathcal{G}_{H,X}$ one can check that

$$\mathcal{G}_{H,X} = \{H + \mathbf{z} : \mathbf{z} \in V(\ell, 2),\ \mathbf{z}M \neq \mathbf{x}_q M \text{ for all } q = 1, \ldots, d\}. \tag{29}$$

By property (b) in Lemma 4.2, $\mathbf{x}_1 M, \ldots, \mathbf{x}_d M$ are nonzero and pairwise distinct. This is equivalent to saying that H avoids X. Thus, from (5) we have $\mu(\Gamma) \leq |G : H| = 2^n$ as claimed in (11). By property (c) in Lemma 4.2, $V(n,2) - \{\mathbf{x}_1 M, \ldots, \mathbf{x}_d M\}$ contains t independent vectors, say $\mathbf{d}_1, \ldots, \mathbf{d}_t$. Since M has rank n, there exist $\mathbf{y}_1, \ldots, \mathbf{y}_t \in V(\ell, 2)$ such that $\mathbf{y}_i M = \mathbf{d}_i$ for each $i = 1, \ldots, t$. Since no $\mathbf{d}_i$ is the same as any $\mathbf{x}_q M$, by (29) we know that all $H + \mathbf{y}_i \in \mathcal{G}_{H,X}$. On the other hand, since $\mathbf{d}_1, \ldots, \mathbf{d}_t$ are independent, $H + \mathbf{y}_1, \ldots, H + \mathbf{y}_t$ are independent in the quotient linear space $V(\ell, 2)/U$. Therefore,

$$|\langle \mathcal{G}_{H,X} \rangle| \geq |\langle H + \mathbf{y}_1, \ldots, H + \mathbf{y}_t \rangle| = 2^t.$$

By (27) and noting $|G : H| = 2^n$ we then have

$$\begin{aligned} \lambda_{j,k}(\Gamma) &\leq 2^n \left(\max\{k, \lceil j/2 \rceil\} + \tfrac{\min\{j-k, \lfloor j/2 \rfloor\}}{|\langle \mathcal{G}_{H,X} \rangle|} \right) - j \\ &\leq 2^n \max\{k, \lceil j/2 \rceil\} + 2^{n-t} \min\{j-k, \lfloor j/2 \rfloor\} - j \end{aligned}$$

as claimed in (10). ◻

The major part of the proof above was to show that the group G contains a subgroup H which avoids X and is such that $|\langle \mathcal{G}_{H,X} \rangle| \geq 2^t$. This was achieved by identifying a matrix M over GF(2) with properties (a)–(c) in Lemma 4.2. From [17, Corollary 4.14], the graph Γ in Theorem 2.5 contains the ℓ-cube Q_ℓ as a spanning subgraph, where ℓ is as in the proof above.

In the case where $\Gamma = Q_d$, we have $\ell = d$, $G = \mathbb{Z}_2^d$, and $Q_d \cong \Gamma(\mathbb{Z}_2^d, X)$, where $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ is the standard basis of $V(d, 2)$. Thus, in the proof of Lemma 4.2, we have $A = I_d$, $Y = N$, and $M = C$, and hence the ith row of M is $\mathbf{x}_i M = \mathbf{c}_i$, for $i = 1, 2, \ldots, d$. Therefore, by Lemma 4.2, in this case we can choose M to be any $d \times n$ matrix over GF(2) with rank n such that its rows are nonzero and pairwise distinct, and the subspace of $V(n, 2)$ spanned by those vectors which are not equal to any row of M has dimension at least t. For each choice of M, the additive group of the null space of M avoids X, and following the proof of Theorem 2.2 we then get an $L(j,k)$-labelling of Q_d which uses 2^n labels and has span $2^n \max\{k, \lceil j/2 \rceil\} + 2^{n-t} \min\{j-k, \lfloor j/2 \rfloor\} - j$.

5. Proof of Theorem 2.9. First, we have the following simple lower bounds for $\lambda_{j,k}(H_{n_1,n_2,\ldots,n_d})$ and $\mu(H_{n_1,n_2,\ldots,n_d})$.

LEMMA 5.1. *Let $n_1 \geq n_2 \geq \cdots \geq n_d$ (≥ 2) be a sequence of $d \geq 2$ integers. Then, for any $j \geq k \geq 1$, we have*

$$\lambda_{j,k}(H_{n_1,n_2,\ldots,n_d}) \geq (n_1 n_2 - 1)k \tag{30}$$

$$\mu(H_{n_1,n_2,\ldots,n_d}) \geq n_1 n_2. \tag{31}$$

Proof. Note that $H_{n_1,n_2,\ldots,n_d}$ contains a subgraph isomorphic to H_{n_1,n_2}. Since H_{n_1,n_2} has diameter 2, under any $L(j,k)$-labelling of $H_{n_1,n_2,\ldots,n_d}$, the $n_1 n_2$ vertices of H_{n_1,n_2} must be assigned labels with a mutual difference of at least k. From this both bounds follow immediately. ◻

Note that, if the equality in (30) occurs, then the equality in (31) occurs as well.

In the proof of Theorem 2.9 we will borrow some ideas from the proof of [9, Theorem 3.1]. However, we do not need a counting argument as used there. We will also use the monotonicity of $\lambda_{j,k}$ and μ: for any subgraph Σ of a graph Γ, we have

$$\lambda_{j,k}(\Sigma) \leq \lambda_{j,k}(\Gamma), \quad \mu(\Sigma) \leq \mu(\Gamma).$$

These hold because any $L(j,k)$-labelling of Γ is also an $L(j,k)$-labelling of Σ as $j \geq k$.

Proof of Theorem 2.9. It suffices to prove

$$\lambda_{j,k}(H_{n_1,n_2,\ldots,n_d}) \leq (n_1n_2-1)\max\{k,\lceil j/2\rceil\} \tag{32}$$

$$\mu(H_{n_1,n_2,\ldots,n_d}) \leq n_1n_2 \tag{33}$$

for any sequence $n_1 \geq n_2 \geq \cdots \geq n_d$ (≥ 2) such that $n_1 > d \geq 2$, n_2 divides n_1, n_i divides n_2 for $i=3,\ldots,d$, and each prime factor of n_i, for $i=1,\ldots,d$, is no less than d. In fact, once this is achieved, then for any sequence $n_1,n_2,\ldots,n_d$ satisfying the conditions of Theorem 2.9 we will have

$$(n_1n_2-1)k \leq \lambda_{j,k}(H_{n_1,n_2,\ldots,n_d}) \leq \lambda_{j,k}(H_{n_1,n_2,\ldots,n_2}) \leq (n_1n_2-1)\max\{k,\lceil j/2\rceil\}$$

and hence (17) and (19) follow. (Note that $\max\{k,\lceil j/2\rceil\}=k$ whenever $2k \geq j$.) Here the first inequality is just (30), the second one is due to the fact that $H_{n_1,n_2,\ldots,n_d}$ is isomorphic to a subgraph of $H_{n_1,n_2,\ldots,n_2}$ and that $\lambda_{j,k}$ is monotonic, and the last one is a special case (where $n_2=n_3=\cdots=n_d$) of (32). The truth of (18) can be proved in a similar way using (31) and (33).

So from now on we suppose that the sequence $n_1 \geq n_2 \geq \cdots \geq n_d \geq 2$ satisfies the conditions in the previous paragraph. Denote $\Gamma := H_{n_1,n_2,\ldots,n_d}$. Then Γ is isomorphic to the Cayley graph $\Gamma(G,X)$, where

$$G := \langle g_1\rangle \times \langle g_2\rangle \times \cdots \times \langle g_d\rangle \tag{34}$$

is the direct product of cyclic groups $\langle g_i\rangle$ of order n_i $(i=1,2,\ldots,d)$ and

$$X := \{(x_1,x_2,\ldots,x_d) : \text{there is exactly one } i \text{ such that } x_i \neq 1\} \tag{35}$$

which is clearly a Cayley set of G. Note that the identity element of G is $1_G = (1,1,\ldots,1)$, where the 1 in the ith position is the identity element of $\langle g_i\rangle$. We will prove the existence of a subgroup H of G such that H avoids X, $|G:H| = n_1n_2$, and $\mathcal{G}_{H,X}$ generates G/H (which is equivalent to saying that $G-HX$ generates G in view of (25)). Once this is achieved, we then have $\lambda_{j,k}(\Gamma) \leq (n_1n_2-1)\max\{k,\lceil j/2\rceil\}$ by (8) and $\mu(\Gamma) \leq n_1n_2$ by (5), and hence (32) and (33) follow.

Since n_2 is a divisor of n_1 and n_i is a divisor of n_2 for $i=3,\ldots,d$, $\langle g_2\rangle$ is isomorphic to a subgroup of $\langle g_1\rangle$, and $\langle g_i\rangle$ is isomorphic to a subgroup of $\langle g_2\rangle$ for $i=3,\ldots,d$. For simplicity of notation, we will take $\langle g_2\rangle$ as a subgroup of $\langle g_1\rangle$, and take each such $\langle g_i\rangle$ as a subgroup of $\langle g_2\rangle$. Thus, for $u=(u_1,u_2,\ldots,u_d) \in G$, we have $\prod_{i=1}^d u_i \in \langle g_1\rangle$, $\prod_{i=2}^d u_i^{i-1} \in \langle g_2\rangle$, and

$$\psi : u \mapsto \left(\prod_{i=1}^d u_i, \prod_{i=2}^d u_i^{i-1}\right)$$

defines a mapping from G to $\langle g_1\rangle \times \langle g_2\rangle$. It is not difficult to check that ψ is a homomorphism from G to $\langle g_1\rangle \times \langle g_2\rangle$. Moreover, ψ is surjective since for any $(u_1,u_2) \in \langle g_1\rangle \times \langle g_2\rangle$ we have $\psi(u_1u_2^{-1},u_2,1,\ldots,1) = (u_1,u_2)$. Define $H := \operatorname{Ker}(\psi)$ to be the kernel of ψ, that is,

$$H = \{u \in G : \psi(u) = (1,1)\}.$$

Then H is a subgroup of G and, by the homomorphism theorem, $G/H \cong \langle g_1 \rangle \times \langle g_2 \rangle$ via the bijection defined by $Hu \leftrightarrow \psi(u)$ for $u \in G$. In particular, H has index $|G : H| = n_1 n_2$ in G. Moreover, we have

Claim 1. H avoids X.

Proof of Claim 1. For any $x = (1, \ldots, x_i, \ldots, 1) \in X$ and $y = (1, \ldots, y_q, \ldots, 1) \in X$, we have $x_i \neq 1$ and $y_q \neq 1$. So $\psi(x) = (x_i, x_i^{i-1}) \neq (1, 1)$, and hence $H \cap X = \emptyset$. Clearly, we have $\psi(xy) = (x_i y_q, x_i^{i-1} y_q^{q-1})$. Thus, if $i = q$, then $\psi(xy) = (1, 1)$ if and only if $xy = (1, 1, \ldots, 1) = 1_G$. If $i \neq q$, say $i < q$, then $\psi(xy) = (1, 1)$ implies $y_q^{q-i} = 1$, which happens only when $d \geq 3$ and the order $o(y_q)$ of y_q is a divisor of $q - i$. In particular, we have $o(y_q) \leq d - 1$ in this case. However, since $o(y_q) > 1$ is a divisor of n_q, we have $o(y_q) \geq d$ by our assumption. This contradiction shows that the product of any two elements of X is not in $H - \{1_G\}$, that is, $H \cap X^2 = \{1_G\}$ and hence claim 1 follows. □

To verify that $\mathcal{G}_{H,X}$ is a generating set of G/H, we prove first the following result, which will be used also in explicitly $L(j, k)$-labelling the vertices of Γ.

Claim 2. There exist $Hv, Hw \in \mathcal{G}_{H,X}$ with orders n_1, n_2, respectively, such that

$$G/H = \langle Hv, Hw \rangle.$$

Proof of Claim 2. To prove this we first assume that $n_1 \neq n_2$. In this case we set $v := (g_1 g_2^{-1}, g_2, 1, \ldots, 1)$ and $w := (g_2^{-1}, g_2, 1, \ldots, 1)$. Then $\psi(v) = (g_1, g_2)$ and $\psi(w) = (1, g_2)$. Clearly, (g_1, g_2) and $(1, g_2)$ generate $\langle g_1 \rangle \times \langle g_2 \rangle$, and they have orders n_1, n_2, respectively. Since $G/H \cong \langle g_1 \rangle \times \langle g_2 \rangle$ via the bijection $Hu \leftrightarrow \psi(u)$ for $u \in G$, it follows that $G/H = \langle Hv, Hw \rangle$ and the orders of Hv, Hw in G/H are n_1, n_2, respectively. Note that, for any $u \in G$, $Hu \cap X \neq \emptyset \Leftrightarrow \psi(u) = \psi(x)$ for some $x \in X$ $\Leftrightarrow \psi(u) = (x_i, x_i^{i-1})$ for some $x_i \neq 1$. In particular, if $Hv \cap X \neq \emptyset$, then $g_1 = x_i$ and $g_2 = x_i^{i-1}$ for some $x_i \neq 1$, which implies $g_2 = g_1^{i-1}$ and hence $i \geq 2$. On the other hand, since $x_i \in \langle g_i \rangle$, it follows from $g_1 = x_i$ that $\langle g_i \rangle = \langle g_1 \rangle$ and hence $n_1 = \cdots = n_i$. In particular, since $i \geq 2$, we have $n_1 = n_2$, which contradicts our assumption. Thus, we must have $Hv \cap X = \emptyset$. Similarly, $Hw \cap X = \emptyset$ for otherwise we would have $(1, g_2) = (x_i, x_i^{i-1})$ for some $x_i \neq 1$, which implies $g_2 = 1$, a contradiction. Therefore, $Hv, Hw \in \mathcal{G}_{H,X}$ and all conditions in claim 2 are satisfied.

In the remaining case we have $n_1 = n_2$, so that g_2 has the same order as g_1. Thus, since $\langle g_2 \rangle$ is a subgroup of $\langle g_1 \rangle$ by our assumption, we have $\langle g_2 \rangle = \langle g_1 \rangle$ and hence $g_1 = g_2^r$ for an integer r, $1 \leq r \leq n_1$, which is coprime to n_1. Set $v := (g_1 g_2^r, g_2^{-r}, 1, \ldots, 1)$ and $w := (g_2^{-1}, g_2, 1, \ldots, 1)$. Then $\psi(v) = (g_1, g_2^{-r}) = (g_1, g_1^{-1})$ and $\psi(w) = (1, g_2)$. By a similar argument as above, one can see that $G/H = \langle Hv, Hw \rangle$ and the orders of Hv, Hw in G/H are n_1, n_2, respectively. Also, $Hw \cap X = \emptyset$ as seen above. If $Hv \cap X \neq \emptyset$, then $g_1 = x_i, g_1^{-1} = x_i^{i-1}$ for some $x_i \neq 1$, and hence $g_1^i = 1$. This implies that n_1 divides i, which is impossible since $1 \leq i \leq d < n_1$. Thus, we must have $Hv \cap X = \emptyset$, and Hv, Hw satisfy the conditions in claim 2. This completes the proof of claim 2. □

Now H avoids X by claim 1, and $\mathcal{G}_{H,X}$ is a generating set of G/H by claim 2. Thus, by (8) we have $\lambda_{j,k}(\Gamma) \leq (n_1 n_2 - 1) \max\{k, \lceil j/2 \rceil\}$ as claimed in (32), and by (5) we have $\mu(\Gamma) \leq n_1 n_2$ as claimed in (33). From our discussion in the first paragraph of this proof, the truth of (17) and (18) follows. Moreover, we can give explicitly an $L(j, k)$-labelling of Γ having span $(n_1 n_2 - 1) \max\{k, \lceil j/2 \rceil\}$ and using $n_1 n_2$ labels. In fact, claim 2 implies that $G/H = \{H(v^i w^\ell) : 0 \leq i < n_1, 0 \leq \ell < n_2\}$. Hence the cosets in G/H can be ordered in the following way to form a sequence. For $1 \leq t \leq n_1 n_2$, there exists a unique pair (i, ℓ) of integers with $1 \leq i \leq n_2$ and

$1 \leq \ell \leq n_1$ such that $t = (i-1)n_1 + \ell$. We then define the tth term Hu_t of the sequence to be $H(v^{\ell-i}w^{i-1})$. It can be checked that, for any two consecutive cosets Hu_t, Hu_{t+1} in the sequence, $H(u_t u_{t+1}^{-1})$ is either Hv^{-1} or Hw^{-1}. Since $Hv \cap X = Hw \cap X = \emptyset$, we have $Hv^{-1} \cap X = Hw^{-1} \cap X = \emptyset$ and hence $H(u_t u_{t+1}^{-1}) \cap X = \emptyset$. From the proof of Corollary 2.4, the labelling under which all elements of Hu_t are labelled by $(t-1)\max\{k, \lceil j/2 \rceil\}$ is an $L(j,k)$-labelling of Γ. This labelling has span $(n_1n_2 - 1)\max\{k, \lceil j/2 \rceil\}$ and uses n_1n_2 labels, and hence is optimal for μ. In the case where $2k \geq j$, we have $\max\{k, \lceil j/2 \rceil\} = k$ and hence (17) together with (30) gives $\lambda_{j,k}(\Gamma) = (n_1n_2 - 1)k$, as stated in (19). Moreover, in this case the $L(j,k)$-labelling above is optimal for $\lambda_{j,k}$ as well. □

Proof of Corollary 2.10. The truth of (20) and (21) follows from (19) and (18), respectively. In addition, in the present case where $2k = j = 2$, the labelling given in the last paragraph of the proof of Theorem 2.9 is a no-hole $L(2,1)$-labelling, and it is optimal for λ and μ simultaneously. □

Remark 5.2. (a) The conditions that $n_1 > d$ and each prime factor of n_1 is no less than d cannot be removed from Theorem 2.9 simultaneously for otherwise the result will not be guaranteed. In fact, for the d-cube Q_d with $d \geq 3$, both conditions are not satisfied; we have $\lambda(Q_d) \geq d + 3$ [19], whilst the right-hand side of (20) is 3. This suggests that hypercubes deserve a different treatment, and this has been done in the previous section.

(b) Unlike [9, Theorem 3.1], Theorem 2.9 and Corollary 2.10 apply even when there are only two complete graph factors (that is, $d = 2$) in the Cartesian product, as long as n_2 divides n_1 and $n_1 > 2$. For such pairs (n_1, n_2), the λ-number of H_{n_1,n_2} is one less than the number of vertices, and each label is used exactly once in any $L(2,1)$-labelling optimal for λ. Harary [14] has asked for a characterization of graphs with this property.

(c) For any graph Γ, we have $\lambda(\Gamma) \geq \mu(\Gamma) - 1$ by definition, and the equality occurs if and only if there exists a no-hole $L(2,1)$-labelling which is optimal for both λ and μ. The Hamming graphs in Corollary 2.10 constitute a family of infinitely many graphs for which $\lambda(\Gamma) = \mu(\Gamma) - 1$ holds.

6. Concluding remarks.

In this paper we introduced a general approach to $L(j,k)$-labelling Cayley graphs on Abelian groups. Then we used this approach to study the $L(j,k)$-labelling problem for Hamming graphs and those graphs whose automorphism groups contain a vertex-transitive Abelian subgroup. The results we obtained for these two families of graphs implied the known results [29, Theorem 3.7] and [9, Theorem 3.1] as special cases. It is expected that the approach would be useful in studying labelling problems for other families of Cayley graphs on Abelian groups.

Based on Theorem 2.9 we may ask naturally the following questions.

Question 6.1. (a) Let j and k be integers with $2k \geq j \geq k \geq 1$. Is

$$\lambda_{j,k}(H_{n_1,n_2,\ldots,n_d}) = (n_1n_2 - 1)k$$

true for any sequence $n_1 \geq n_2 \geq \cdots \geq n_d$ of $d \geq 2$ integers which are no less than 2 but not all equal to 2?

(b) In particular, is $\lambda(H_{n_1,n_2,\ldots,n_d}) = n_1n_2 - 1$ true for the same sequence?

In other words, we would like to know whether (19) is valid for any Hamming graph other than a hypercube provided that $2k \geq j$. The result in [10, Theorem 4.2] shows that the answer to (b) is affirmative for H_{n_1,n_2} with $2 \leq n_2 \leq n_1$ and $(n_1, n_2) \neq (2,2)$. In general, a recent result of the author with Chang and Lu [5] shows that, if n_1 is substantially larger than n_2 and d, then the answer to (b) above is affirmative. As we have seen in the proof of Theorem 2.9, if we could find a subgroup H of the group G (defined in (34)) such that H avoids the Cayley set X (defined in (35)), $|G : H| = n_1 n_2$ and $\mathcal{G}_{H,X}$ is a generating set of G/H, then the answer to both (a) and (b) of Question 6.1 is positive. However, we suspect that in general the answers to these questions are negative.

Acknowledgments. The author appreciates an anonymous referee for his/her suggestions which led to better structure of this paper. He also thanks Professor Gerard J. Chang for his comments which led to an improved presentation of Theorem 2.9, and Dr. Changhong Lu for his help in sorting out the lambda number of a certain special graph.

REFERENCES

[1] N. L. Biggs, *Algebraic graph theory* (2nd edition), Cambridge University Press, Cambridge, UK, 1993.

[2] G. J. Chang, W-T. Ke, D. Kuo, D. D-F. Liu, and R. K. Yeh, *On $L(d,1)$-labelings of graphs*, Discrete Math., 220 (2000), pp. 57–66.

[3] G. J. Chang and D. Kuo, *The $L(2,1)$-labeling problem on graphs*, SIAM J. Discrete Math., 9 (1996), pp. 309–316.

[4] G. J. Chang and C. Lu, *Distance-two labelings of graphs*, European J. Combin., 24 (2003), pp. 53–58.

[5] G. J. Chang, C. Lu, and S. Zhou, *Minimum spans of Hamming graphs under distance-two labelling*, in preparation.

[6] D. J. Erwin, J. P. Georges, and D. W. Mauro, *On labeling the vertices of products of complete graphs with distance constraints*, Naval Res. Logist., 52 (2005), pp. 138–141.

[7] J. P. Georges and D. W. Mauro, *Generalized vertex labelings with a condition at distance two*, Congr. Numer., 109 (1995), pp. 141–159.

[8] J. P. Georges and D. W. Mauro, *Some results on λ^j_k-numbers of the products of complete graphs*, Congr. Numer., 140 (1999), pp. 141–160.

[9] J. P. Georges, D. W. Mauro, and M. I. Stein, *Labeling products of complete graphs with a condition at distance two*, SIAM J. Discrete Math., 14 (2000), pp. 28–35.

[10] J. P. Georges, D. W. Mauro, and M. A. Whittlesey, *Relating path coverings to vertex labellings with a condition at distance two*, Discrete Math., 135 (1994), pp. 103–111.

[11] J. R. Griggs and R. K. Yeh, *Labelling graphs with a condition at distance two*, SIAM J. Discrete Math., 5 (1992), pp. 586–595.

[12] J. L. Gross, *Every connected regular graph of even degree is a Schreier coset graph*, J. Combinatorial Theory Ser. B, 22 (1977), pp. 227–232.

[13] W. K. Hale, *Frequency assignment: Theory and applications*, Proc. IEEE, 68 (1080), pp. 1497–1514.

[14] F. Harary, *Coloring costs of a graph and the radio coloring number*, private communication.

[15] F. Harary and M. Plantholt, *Graphs whose radio coloring number equals the number of nodes*, in Graph colouring and Applications (Montréal, QC, 1997), CRM Proc. Lecture Notes 23, AMS, Providence, RI, 1999, pp. 99–100.

[16] W. Imrich, *Graphs with transitive abelian automorphism group*, in Combin. Theory Appl., Vol. 4, Colloq. Math. Soc. János Bolyai, North-Holland, Amsterdam, 1970, pp. 651–656.

[17] W. Imrich and S. Klavžar, *Product Graphs*, Wiley-Interscience, New York, 2000.

[18] W. Imrich and M. E. Watkins, *On automorphism groups of Cayley graphs*, Period. Math. Hungar., 7 (1976), pp. 243–258.

[19] K. Jonas, *Graph coloring analogues with a condition at distance two: $L(2,1)$-labellings and listed λ-labellings*, Ph.D. thesis, Department of Mathematics, University of South Carolina, Columbia, SC, 1993.

[20] D. Král and R. Skrekovski, *A theorem about the channel assignment problem*, SIAM J. Discrete Math., 16 (2003), pp. 426–437.

[21] D. Marŭsič, *Hamiltonian circuits in Cayley graphs*, Discrete Math., 46 (1983), pp. 49–54.

[22] F. S. Roberts, *T-colorings of graphs: Recent results and open problems*, Discrete Math., 93 (1991), pp. 229–245.

[23] F. S. Roberts, *No-hole 2-distant colorings*, in Graph-theoretic Models in Computer Science II (Las Cruces, NM, 1988–1990), Math. Comput. Modelling, 17 (1993), pp. 139–144.

[24] D. Sakai, *Labelling chordal graphs: Distance two condition*, SIAM J. Discrete Math., 7 (1994), pp. 133–140.

[25] J. S. Rose, *A Course on Group Theory*, Cambridge University Press, Cambridge, UK, 1978.

[26] A. W. To, *personal communication*, 2003.

[27] T. Walsh, *The cost of radio-colouring paths and cycles*, in Graph Colouring and Applications (Montréal, QC, 1997), CRM Proc. Lecture Notes 23, AMS, Providence, RI, 1999, pp. 131–133.

[28] P.-J. Wan, *Near-optimal conflict-free channel set assignments for an optical cluster-based hypercube network*, J. Comb. Optim., 1 (1997), pp. 179–186.

[29] M. A. Whittlesey, J. P. Georges, and D. W. Mauro, *On the λ-number of Q_n and related graphs*, SIAM J. Discrete Math., 8 (1995), pp. 499–506.

[30] R. K. Yeh, *Labelling graphs with a condition at distance two*, Ph.D. thesis, Department of Mathematics, University of South Carolina, Columbia, SC, 1990.

SIAM J. Discrete Math.
Vol. 19, No. 4, pp. 1004–1010

THE DISCRETE SINE TRANSFORM AND THE SPECTRUM OF THE FINITE q-ARY TREE*

FABIO SCARABOTTI†

Abstract. Recently, He, Liu, and Strang [*Stud. Appl. Math.*, 110 (2003), pp. 123–138] have computed the spectrum of the adjacency matrix of a class of finite trees. In this paper, we propose a different method and apply it to the slightly different class of finite q-ary trees.

Key words. tree, spectrum, discrete sine transform, Radon transform

AMS subject classifications. 05C50, 43A85, 44A12

DOI. 10.1137/S0895480104445344

1. Introduction.

In [6], He, Liu, and Strang computed the spectrum of the finite trees that can be obtained by taking a ball of finite radius in an infinite homogeneous tree. These trees are rooted, all the leaves (end points) have the same distance from the root, and all the internal vertices have the same degree. Their method is based on a factorization of the characteristic polynomial obtained through a recursion on the diameter of the tree.

In the present paper, we deal with a slightly different kind of tree: the q-ary tree of height n. This means that we have a root which has q sons, q^2 grandsons, etc., for n generations; in this case the root has degree q, while all other internal vertices have degree $q+1$. For these trees we propose a method that is based on a preliminary decomposition of the space of all complex valued functions defined on the vertex set of the tree.

On each level of the tree, we use the decomposition into irreducible representations of the group of automorphisms of the tree $\mathrm{Aut}(T)$ [5], [7]. But note that our proof is very elementary: no knowledge of representation theory is required, only some elementary linear algebra. We obtain a decomposition by means of suitable Radon transforms that intertwine the representations on the various levels of the tree. They are strictly connected with the adjacency operator and the geometry of the tree. To get the spectrum, we apply the discrete sine transform to the action of the adjacency operator on such a decomposition.

Our method has a close resemblance to the proof of a theorem of Stanley [9, Theorem 4.14].

2. The tree and its adjacency operator.

A *tree* T is a connected graph without circuits. We say that T is *rooted* if it has a distinguished vertex x_0, called the root. We say that T is *q-ary* of *height n* if it satisfies the following three conditions: the root has degree q; a vertex is a leaf (i.e., it has degree 1) if and only if its distance from the root is equal to n; all the remaining vertices have degree $q+1$. Figure 1 is the ternary tree of height 3. In what follows, T will be a q-ary tree of height n. We will identify T with the set of all its vertices, and we will write $x \sim y$ to denote that $x, y \in T$ are *adjacent*, i.e., they are connected by an edge. We will denote by Ω_k the set

*Received by the editors July 26, 2004; accepted for publication (in revised form) July 28, 2005; published electronically January 26, 2006.

http://www.siam.org/journals/sidma/19-4/44534.html

†Dipartimento MeMoMat, Università degli Studi di Roma “La Sapienza,” via A. Scarpa 8, 00161 Roma Italy (scarabot@dmmm.uniroma1.it).

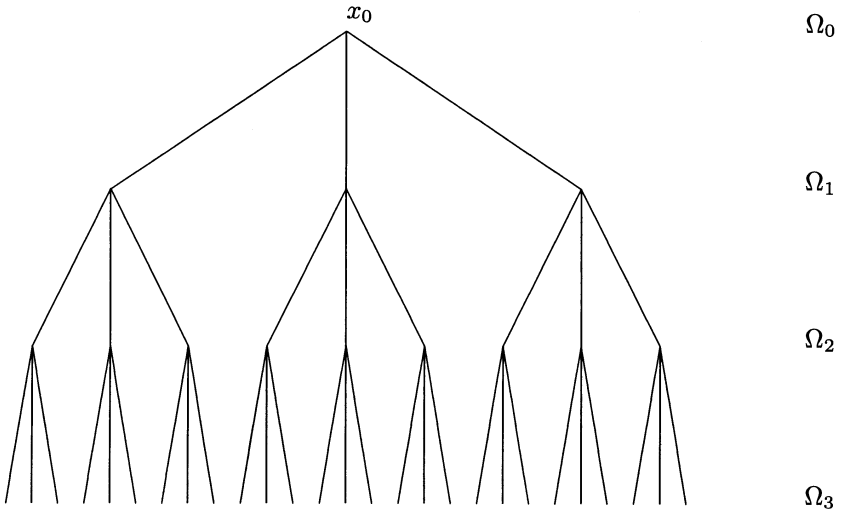


FIG. 1.

of vertices whose distance from the root is equal to k, $k = 0, 1, \ldots, n$ (the *k-level* of the tree). When $x \sim y$ and x belongs to a higher level than y, e.g., $x \in \Omega_k$ and $y \in \Omega_{k+1}$, we will say that x is the *father* of y and that y is a *son* of x, and we will write $x \succ y$. The space $\{f : T \to \mathbb{C}\}$ of all complex valued functions defined on T will be denoted by $L(T)$; it will be endowed with the scalar product $\langle f_1, f_2\rangle = \sum_{x\in T} f_1(x)\overline{f_2(x)}$. The *adjacency operator* A of T is defined by setting $(Af)(x) = \sum_{y\in T: x\sim y} f(y)$ for all $x \in T$ and $f \in L(T)$. By definition [2], the spectrum of the tree coincides with the spectrum of its adjacency operator A.

3. The discrete sine transform and the spectrum of the path. Let B_n be the $n \times n$ tridiagonal matrix

$$B_n = \begin{pmatrix} 0 & 1 & & & & & \\ 1 & 0 & 1 & & & & \\ & 1 & 0 & 1 & & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & 1 & 0 & 1 \\ & & & & & 1 & 0 \end{pmatrix}.$$

Set $\alpha = \frac{\pi}{n+1}$. Then the $n \times n$ matrix

$$S_n = \sqrt{\frac{2}{n+1}} \begin{pmatrix} \sin\alpha & \sin 2\alpha & \ldots & \sin(n-1)\alpha & \sin n\alpha \\ \sin 2\alpha & \sin 4\alpha & \ldots & \sin 2(n-1)\alpha & \sin 2n\alpha \\ \vdots & \vdots & & \vdots & \vdots \\ \sin(n-1)\alpha & \sin 2(n-1)\alpha & \ldots & \sin(n-1)^2\alpha & \sin n(n-1)\alpha \\ \sin n\alpha & \sin 2n\alpha & \ldots & \sin(n-1)n\alpha & \sin n^2\alpha \end{pmatrix}$$

is symmetric and orthogonal and diagonalizes B_n:

$$S_nB_nS_n = \begin{pmatrix} 2\cos\alpha & & & \\ & 2\cos 2\alpha & & \\ & & \ddots & \\ & & & 2\cos n\alpha \end{pmatrix}. \tag{1}$$

This is the *discrete sine transform* (DST) [10]. Moreover, (1) is the computation of the spectrum of the tree T in the case $q = 1$ (the path): B_n is the matrix representing the adjacency operator of the path if we take the standard basis $\{\delta_x : x \in T\}$ for $L(T)$, where $\delta_x(y) = 1$ if $x = y$, $\delta_x(y) = 0$ if $x \neq y$.

Remark. The characteristic polynomial $\det(\lambda I - B_n)$ of B_n, also called the characteristic polynomial of the path, may be expressed by the *Chebyshev polynomials of the second kind* [2, p. 11]: $\det(\lambda I - B_n) = U_n(\lambda/2)$. The computation of the spectrum of the tree in [6] is based in a factorization of the characteristic polynomial of the tree in terms of (rescaled) Chebyshev polynomials of the second kind: in the notations of [6], $p_n(\lambda) = (k-1)^{n/2}U_n(\frac{\lambda}{2\sqrt{k-1}})$; see also [3, section 1.4].

4. The Radon transforms $\boldsymbol{R}$ and $\boldsymbol{R^*}$. First note that $T = \sqcup_{k=0}^n \Omega_k$ (where $\sqcup$ denotes a disjoint union) leads to the orthogonal decomposition $L(T) = \oplus_{k=0}^n L(\Omega_k)$. Then we define the linear operator $R : \oplus_{k=1}^n L(\Omega_k) \to \oplus_{k=0}^{n-1} L(\Omega_k)$ by setting

$$(Rf)(x) = \sum_{y \in T : y \prec x} f(y)$$

for every $f \in \oplus_{k=1}^n L(\Omega_k)$ and $x \in \sqcup_{k=0}^{n-1} \Omega_k$. In other words, the value of Rf on x is the sum of the values of f on the sons of x. The adjoint of R is the linear operator $R^* : \oplus_{k=0}^{n-1} L(\Omega_k) \to \oplus_{k=1}^n L(\Omega_k)$ given by

$$(R^*f)(x) = f(y), \qquad \text{where } y \text{ is the father of } x,$$

for every $f \in \oplus_{k=0}^{n-1} L(\Omega_k)$ and $x \in \sqcup_{k=1}^n \Omega_k$.

Clearly R is surjective and R^* is injective. Moreover, R maps $L(\Omega_k)$ onto $L(\Omega_{k-1})$ and R^* maps $L(\Omega_{k-1})$ into $L(\Omega_k)$, $k = 1, 2, \ldots, n$. In particular, $(R^*)^{k-h}(L(\Omega_h))$ is a homomorphic image of $L(\Omega_h)$ in $L(\Omega_k)$: it consists of all functions in $L(\Omega_k)$ that are constant on the leaves of each q-ary subtree of T of height $k - h$ rooted on a vertex in Ω_h.

We also define $W_k = L(\Omega_k) \cap \ker R$, $k = 1, 2, \ldots, n$ and $W_0 = L(\Omega_0) \equiv \mathbb{C}$. Note that $\dim W_0 = 1$ and that $\dim W_k = q^k - q^{k-1}$.

The following identity is easy but important:

$$RR^*f = qf. \tag{2}$$

Indeed, $(RR^*f)(x) = \sum_{y \in T : y \prec x} (R^*f)(y) = qf(x)$.

LEMMA 4.1. *For $k = 1, 2, \ldots, n$ we have an orthogonal decomposition of $L(\Omega_k)$:*

$$L(\Omega_k) = (R^*)^k(W_0) \oplus (R^*)^{k-1}(W_1) \oplus \cdots \oplus (R^*)(W_{k-1}) \oplus W_k.$$

Proof. First note that a consequence of (2) is that

$$\langle R^*f_1, R^*f_2 \rangle = q\langle f_1, f_2 \rangle, \tag{3}$$

and this is also easy to prove directly.

Using (3), we can iterate the decomposition $L(\Omega_k) = R^*(L(\Omega_{k-1})) \oplus [\ker R \cap L(\Omega_k)] \equiv R^*(L(\Omega_{k-1})) \oplus W_k$:

$$\begin{aligned} L(\Omega_k) =& R^*(L(\Omega_{k-1})) \oplus W_k \\ =& (R^*)^2(L(\Omega_{k-2})) \oplus R^*(W_{k-1}) \oplus W_k \\ & \dots \\ =& (R^*)^k(W_0) \oplus (R^*)^{k-1}(W_1) \oplus \dots \oplus (R^*)(W_{k-1}) \oplus W_k. \qquad \square \end{aligned}$$

In other words, $(R^*)^k(W_0)$ is the space of constant functions on Ω_k and $(R^*)^{k-h}(W_h)$ is the space of all functions in $L(\Omega_k)$ that are constant on the leaves of each q-ary subtree of T of height $k-h$ rooted on a vertex in Ω_h and whose sum on the leaves of every q-ary subtree of height $k-h+1$ rooted on a vertex in Ω_{h-1} is equal to zero.

Another fundamental identity relates the adjacency operator A to the Radon transforms R and R^*: if $f \in L(T)$ and $f = f_0 + f_1 + \dots + f_n$ with $f_h \in L(\Omega_h)$, then

$$Af = Rf_1 + \sum_{h=1}^{n-1}(R^* f_{h-1} + Rf_{h+1}) + R^* f_{n-1}, \tag{4}$$

where $Rf_1 \in L(\Omega_0)$, $R^* f_{h-1} + Rf_{h+1} \in L(\Omega_h)$, and $R^* f_{n-1} \in L(\Omega_n)$. For instance, if $x \in \Omega_h$ with $1 \le h \le n-1$, then

$$\begin{aligned} (Af)(x) = \sum_{y \sim x} f(y) &= \sum_{z \in \Omega_{h+1} : z \sim x} f(z) + \sum_{y \in \Omega_{h-1} : y \sim x} f(y) \\ &= (Rf)(x) + (R^* f)(y) \equiv (Rf_{h-1})(x) + (R^* f_{h+1})(x). \end{aligned}$$

Remarks. (1) We call R and R^* Radon transforms because they are (natural) operators intertwining $L(\Omega_k)$ and $L(\Omega_{k+1})$ as permutation representations of $\mathrm{Aut}(T)$, the group of automorphisms of T; see [8]. The decomposition in Lemma 4.1 is well known and coincides with the decomposition of $L(\Omega_k)$ into irreducible representations of $\mathrm{Aut}(T)$; see [5], [7], and also [1, pp. 152–156], which has a more algebraic form. But in our case we are not on a homogeneous space: $\mathrm{Aut}(T)$ does not act transitively on T. Therefore we may not apply the finite Fourier transform (for which we refer to [4]) to get the spectrum of T. Nevertheless, A is $\mathrm{Aut}(T)$-invariant, and therefore the eigenspaces of A must be direct sums of irreducible representations of $\mathrm{Aut}(T)$, as we will show in the next section.

(2) The operators R^* and R can also be seen as instances of "up" and "down" operators as in [9] (but note that Stanley would draw the tree with the root at the bottom and the leaves at the top; therefore in his terminology R goes down and R^* goes up). However, our tree is not a differential poset of Stanley: it is easy to see that in our case

$$(RR^* - R^*R)f = \begin{cases} qf & \text{if} \quad f \in W_k, \\ 0 & \text{if} \quad f \in L(\Omega_k), f \perp W_k, \end{cases}$$

while the definition of differential poset requires that the commutator $RR^* - R^*R$ is always a multiple of the identity. Nevertheless, our computation of the spectrum of the tree in the following section has a close resemblance to the proof of Theorem 4.14 in [9].

5. The spectrum of the tree.

LEMMA 5.1. *For $k = 0, 1, \dots, n$ and $l = 1, 2, \dots, n-k+1$ set*

$$W_{k,l} = \left\{ \sum_{h=0}^{n-k} \frac{1}{q^{h/2}} \sin \frac{(h+1)l\pi}{n-k+2} \cdot f \quad : \quad f \in W_k \right\}.$$

Then each $W_{k,l}$ is an eigenspace of A. The corresponding eigenvalue is equal to $2\sqrt{q}\cos\frac{\pi l}{n-k+2}$ and $\oplus_{h=0}^{n-k}(R^*)^h W_k = \oplus_{l=1}^{n-k+1} W_{k,l}$.

Proof. If $f \in W_k$ and $a_0, a_1, \dots, a_{n-k} \in \mathbb{C}$, then from (2) and (4) it follows that

$$\begin{aligned}
&A(a_0 f + a_1 R^* f + \cdots + a_{n-k}(R^*)^{n-k} f) \\
&= a_0 Rf + a_1 RR^* f + \sum_{h=k+1}^{n-1} \left[a_{h-k-1} R^*(R^*)^{h-k-1} f + a_{h-k+1} R(R^*)^{h-k+1} f \right] \\
&\quad + a_{n-k-1} R^*(R^*)^{n-k-1} f \\
&= a_1 q f + \sum_{h=k+1}^{n-1} \left[a_{h-k-1} + q a_{h-k+1} \right] (R^*)^{h-k} f + a_{n-k-1}(R^*)^{n-k} f.
\end{aligned}$$

Therefore $F = a_0 f + a_1 R^* f + \cdots + a_{n-k}(R^*)^{n-k} f$ is an eigenvector of A; i.e., $AF = \lambda F$ if and only if the coefficients $a_0, a_1, \dots, a_{n-k}$ solve the eigenvalue problem

$$\left\{ \begin{array}{lll} a_{h-1} + q a_{h+1} = \lambda a_h & \text{for} & h = 1, 2, \dots, n-k-1, \\ q a_1 = \lambda a_0; \quad a_{n-k-1} = \lambda a_{n-k}. & & \end{array} \right. \tag{5}$$

With the substitutions $b_h = q^{h/2} a_h$, $h = 0, 1, \dots, n-k$, and $\mu = \frac{\lambda}{\sqrt{q}}$ (5) becomes

$$\left\{ \begin{array}{lll} b_{h-1} + b_{h+1} = \mu b_h & \text{for} & h = 1, 2, \dots, n-k-1, \\ b_1 = \mu b_0; \quad b_{n-k-1} = \mu b_{n-k}, & & \end{array} \right.$$

which is the eigenvalue problem solved by the DST. Therefore from section 3 one recovers the eigenvalues and the eigenspaces in the statement. Finally, $\oplus_{h=0}^{n-k}(R^*)^h W_k \equiv \{a_0 f + a_1 R^* f + \cdots + a_{n-k}(R^*)^{n-k} f : f \in W_k,\ a_0, a_1, \dots, a_{n-k} \in \mathbb{C}\}$ is clearly equal to $\oplus_{l=1}^{n-k+1} W_{k,l}$, because the rows of the matrix of the DST form an orthogonal basis. □

Now we can state and prove the main theorem on the spectral analysis of A. We will write $(a, b) = 1$ to indicate that the integers a and b are relatively prime.

THEOREM 5.2.

1. *The spectrum of A coincides with the set $\{2\sqrt{q}\cos\frac{\pi l}{n-k+2} : k = 0, 1, \dots, n;\ l = 1, 2, \dots, n-k+1; (l, n-k+2) = 1\}$.*
2. *Suppose that $0 \le k \le n$, $1 \le l \le n-k+1$, and $(l, n-k+2) = 1$. If $k = (n-k+2)s + r$, with $0 \le r \le n-k+1$, then the eigenspace corresponding to $2\sqrt{q}\cos\frac{\pi l}{n-k+2}$ is*

$$\oplus_{t=0}^{s} W_{k-t(n-k+2),l(t+1)}.$$

3. *The multiplicity of $2\sqrt{q}\cos\frac{\pi l}{n-k+2}$ is equal to*

$$\begin{array}{ll} (q^r - q^{r-1})\dfrac{q^{(n-k+2)(s+1)} - 1}{q^{n-k+2} - 1} & \textit{if} \quad 1 \le r \le n-k+1, \\ 1 + (q^{n-k+2} - q^{n-k+1})\dfrac{q^{(n-k+2)s} - 1}{q^{n-k+2} - 1} & \textit{if} \quad r = 0. \end{array}$$

Proof. From the decomposition $L(T) = \oplus_{k=0}^{n} L(\Omega_k)$ and Lemmas 4.1 and 5.1 we have

$$L(T) = \oplus_{k=0}^{n} \oplus_{h=0}^{n-k} (R^*)^h W_k = \oplus_{k=0}^{n} \oplus_{l=1}^{n-k+1} W_{k,l},$$

and therefore Lemma 5.1 yields part 1. To prove part 2, observe first that $k = s(n-k+2)+r$ is equivalent to $n+2 = (s+1)(n-k+2)+r$. Therefore $(t+1)(n-k+2)$ is equal to $n-k_1+2$ with $0 \le k_1 \le n$ if and only if $k_1 = k - t(n-k+2)$ with $0 \le t \le s$. Exactly for those values of t the eigenvalue $2\sqrt{q}\cos\frac{l\pi}{n-k+2}$ appears again in the form $2\sqrt{q}\cos\frac{l(t+1)\pi}{(t+1)(n-k+2)}$, and the corresponding eigenspace is $W_{k-t(n-k+2),l(t+1)}$. Moreover, $\sum_{t=0}^{s} \dim W_{k-t(n-k+2),l(t+1)}$ is equal to

$$\begin{aligned}\sum_{t=0}^{s}&(q^{k-t(n-k+2)} - q^{k-t(n-k+2)-1})\\ &= (q^r - q^{r-1})\sum_{w=0}^{s} q^{(n-k+2)w} = (q^r - q^{r-1})\frac{q^{(n-k+2)(s+1)}-1}{q^{n-k+2}-1} \qquad \text{for} \quad r \ge 1,\end{aligned}$$

$$\begin{aligned}1 + \sum_{t=0}^{s-1}(q^{k-t(n-k+2)} - q^{k-t(n-k+2)-1}) &= 1 + (q^{n-k+2} - q^{n-k+1})\sum_{w=0}^{s-1} q^{(n-k+2)w}\\ = 1 + (q^{n-k+2} - q^{n-k+1})\frac{q^{(n-k+2)s}-1}{q^{n-k+2}-1} &\qquad \text{for} \quad r = 0. \qquad \square\end{aligned}$$

Our method might be applied to other classes of rooted trees. For instance, consider a tree where each vertex at level k has q_k sons, $k = 0, 1, \dots, n-1$. In this case (5) is replaced by the more general eigenvalue problem

$$(6) \qquad \begin{cases} a_{h-1} + q_{k+h}a_{h+1} = \lambda a_h & \text{for} \quad h = 1, 2, \dots, n-k-1,\\ q_k a_1 = \lambda a_0; \quad a_{n-k-1} = \lambda a_{n-k}. \end{cases}$$

In general, this problem does not have an explicit elementary solution. Nevertheless, in particular cases some of the eigenvalues are computable. For $q_0 = q+1$ and $q_k = q$, $k = 1, 2, \dots, n$ we obtain the trees in [6], and in this case almost all eigenvalues are computable; those missing correspond to the subspace $\oplus_{h=0}^{n}(R^*)^h(W_0)$: now for $k \ge 1$ (6) reduces to (5).

Acknowledgment. I express my warm gratitude to Professor Gilbert Strang for his remarks and encouragement.

REFERENCES

[1] H. Bass, M. V. Otero-Espinar, D. Rockmore, and C. Tresser, *Cyclic Renormalization and Automorphism Groups of Rooted Trees*, Lecture Notes in Math. 1621, Springer-Verlag, New York, 1996.

[2] N. Biggs, *Algebraic Graph Theory*, 2nd ed., Cambridge Math. Lib., Cambridge University Press, Cambridge, UK, 1993.

[3] G. Davidoff, P. Sarnak, and A. Valette, *Elementary Number Theory, Group Theory and Ramanujan Graphs*, London Math. Soc. Stud. Texts 55, Cambridge University Press, Cambridge, UK, 2003.

[4] P. Diaconis, *Group Representations in Probability and Statistics*, Institute of Mathematical Statistics Lecture Notes—Monograph Series, 11. Institute of Mathematical Statistics, Hayward, CA, 1988.

[5] A. Figà-Talamanca, *An application of Gelfand pairs to a problem of diffusion in compact ultrametric spaces*, in Topics in Probability and Lie Groups: Boundary Theory, CRM Proc. Lecture Notes 28, AMS, Providence, RI, 2001, pp. 51–67.

[6] L. He, X. Liu, and G. Strang, *Trees with Cantor eigenvalue distribution*, Stud. Appl. Math., 110 (2003), pp. 123–138.

[7] G. Letac, *Les fonctions sphériques d'un couple de Gelfand symétrique et les chaînes de Markov*, Adv. Appl. Probab., 14 (1982), pp. 272–294.

[8] F. Scarabotti, *Fourier analysis of a class of finite Radon transforms*, SIAM J. Discrete Math., 16 (2003), pp. 545–554.

[9] R. P. Stanley, *Differential posets*, J. Amer. Math. Soc., 1 (1988), pp. 919–961.

[10] G. Strang, *The discrete cosine transform*, SIAM Rev., 41 (1999), pp. 135–147.

SIAM J. Discrete Math.
Vol. 19, No. 4, pp. 1011–1028

IMPROVED p-ARY CODES AND SEQUENCE FAMILIES FROM GALOIS RINGS OF CHARACTERISTIC p^2*

SAN LING† AND FERRUH ÖZBUDAK‡

Abstract. This paper explores the applications of a recent bound on some Weil-type exponential sums over Galois rings in the construction of codes and sequences. A family of codes over $\mathbb{F}_p$, mostly nonlinear, of length p^{m+1} and size $p^2 \cdot p^{m(D-\lfloor D/p^2 \rfloor)}$, where $1 \le D \le p^{m/2}$, is obtained. The bound on this type of exponential sums provides a lower bound for the minimum distance of these codes. Several families of pairwise cyclically distinct p-ary sequences of period $p(p^m-1)$ of low correlation are also constructed. They compare favorably with certain known p-ary sequences of period p^m-1. Even in the case $p=2$, one of these families is slightly larger than the family $Q(D)$ in section 8.8 in [T. Helleseth and P. V. Kumar, *Handbook of Coding Theory*, Vol. 2, North-Holland, 1998, pp. 1765–1853], while they share the same period and the same bound for the maximum nontrivial correlation.

Key words. Galois rings, p-ary code, exponential sum, p-ary sequence

AMS subject classifications. 94B05, 94B40, 11T23

DOI. 10.1137/S089548010444506X

1. Introduction.

Bounds on exponential sums over finite fields, such as the Weil–Carlitz–Uchiyama bound, have been found to be useful in applications such as coding theory and sequence designs. The analogue of the Weil–Carlitz–Uchiyama bound for Galois rings was presented in [K-H-C]. An improved bound for a related Weil-type exponential sum over Galois rings of characteristic 4, which is also sometimes called the trace of exponential sums, was obtained in [H-K-M-S] and was used in [S-K-H] to construct a family of binary codes with the same length and size as the Delsarte–Goethals codes, but whose minimum distance is significantly bigger. The shortening of these codes also leads to efficient binary sequences.

Recently, an analogue of the bound of [H-K-M-S] was obtained for Galois rings of characteristic p^2, for all primes p [L-O]. In this paper, we explore some applications of this bound to the construction of codes and sequences. Starting from some trace codes over $\mathbb{Z}_{p^2}$ and applying the Gray map, a family of codes over $\mathbb{F}_p$ of length p^{m+1} and size $p^2 \cdot p^{m(D-\lfloor D/p^2 \rfloor)}$, where $1 \le D \le p^{m/2}$, is constructed. This family is a generalization of the family of binary codes of [S-K-H] and it is a family of nonlinear codes in general. A lower bound for their minimum distance is obtained through the bound of [L-O]. Using the generalized Nechaev–Gray map, several families of

*Received by the editors July 9, 2004; accepted for publication (in revised form) March 29, 2005; published electronically February 3, 2006.
http://www.siam.org/journals/sidma/19-4/44506.html

†Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543, Republic of Singapore (matlings@nus.edu.sg). Current address: Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Block 5 Level 3, 1 Nanyang Walk, Singapore 637616, Republic of Singapore (lingsan@ntu.edu.sg). The research of this author was partially supported by NUS-ARF research grant R-146-000-029-112 and by DSTA research grant POD0411403.

‡Department of Mathematics and Institute of Applied Mathematics, Middle East Technical University, İnönü Bulvarı, 06531, Ankara, Turkey (ozbudak@math.metu.edu.tr). The research of this author was partially supported by the Turkish Academy of Sciences in the framework of the Young Scientists Award Programme (F.Ö./TÜBA-GEBIP/2003-13).

pairwise cyclically distinct p-ary sequences of period $p(p^m-1)$ of low correlation are also obtained. They compare favorably with certain known p-ary sequences of period p^m-1 (cf. [H-K, Table 4]). In fact, even in the case $p=2$, one of these families is slightly larger than the family $Q(D)$ of [H-K, section 8.8], while they share the same period and the same bound for the maximum nontrivial correlation.

We fix the following conventions throughout the paper:

- p: a prime number,
- m: an integer with $m \geq 2$,
- $\mathbb{F}_p$, $\mathbb{F}_{p^m}$: finite fields of cardinality p and p^m,
- $\mathrm{tr}_m : \mathbb{F}_{p^m} \to \mathbb{F}_p$: the trace map from $\mathbb{F}_{p^m}$ onto $\mathbb{F}_p$,
- $\mathrm{GR}(p^2, m)$: a Galois ring of characteristic p^2 with cardinality p^{2m},
- $\mathbb{Z}_{p^2}$: the ring of integers modulo p^2,
- $\mathrm{Tr}_m : \mathrm{GR}(p^2, m) \to \mathbb{Z}_{p^2}$: the trace map from $\mathrm{GR}(p^2, m)$ onto $\mathbb{Z}_{p^2}$,
- Γ_m: the Teichmüller set in $\mathrm{GR}(p^2, m)$,
- β: a primitive (p^m-1)th root of unity in $\mathrm{GR}(p^2, m)$,
- $\rho : \mathrm{GR}(p^2, m) \to \mathrm{GR}(p^2, m)/p\mathrm{GR}(p^2, m) \cong \mathbb{F}_{p^m}$: reduction modulo p map in $\mathrm{GR}(p^2, m)$,
- $\omega = \rho(\beta)$: a primitive (p^m-1)th root of unity in $\mathbb{F}_{p^m}$.

We extend ρ to the polynomial ring mapping $\rho : \mathrm{GR}(p^2, m)[x] \to \mathbb{F}_{p^m}[x]$ by its action on the coefficients. Note that the restricted map $\rho_{|_{\Gamma_m[x]}} : \Gamma_m[x] \to \mathbb{F}_{p^m}[x]$ is one-to-one and onto.

We recall that the Frobenius operator Frob on $\mathrm{GR}(p^2, m)$ is defined as

$$\mathrm{Frob}(a+pb) = a^p + pb^p, \quad \text{where } a, b \in \Gamma_m.$$

Moreover Frob is extended to $\mathrm{GR}(p^2, m)[x]$ as

$$\mathrm{Frob}\left(\sum_{i=1}^{l} a_i x^i\right) = \sum_{i=1}^{l} \mathrm{Frob}(a_i) x^{pi}.$$

A polynomial $f(x) \in \mathrm{GR}(p^2, m)[x]$ is called *nondegenerate* if it cannot be written in the form

$$f(x) = \mathrm{Frob}(g(x)) - g(x) + u \bmod p^2,$$

where $g(x) \in \mathrm{GR}(p^2, m)[x]$ and $u \in \mathrm{GR}(p^2, m)$.

2. $\mathbb{Z}_{p^2}$-linear trace codes.

In this section we construct a family of codes over $\mathbb{F}_p$ starting from some trace codes over $\mathbb{Z}_{p^2}$ and applying the Gray map. These codes can be considered as p-ary version of the codes of [S-K-H]. We obtain a lower bound for their minimum distance using a bound of [L-O].

We begin with a definition.

DEFINITION 2.1. *For a finite $\mathbb{Z}_{p^2}$-module $S \subseteq \mathrm{GR}(p^2, m)[x]$, we define the subsets S_0, $S_1 \subseteq \Gamma_m[x]$ as*

$$S_0 = \{a(x) \in \Gamma_m[x] : \textit{there exists } b(x) \in \Gamma_m[x] \textit{ such that } a(x) + pb(x) \in S\} \quad \textit{and}$$

$$S_1 = \{b(x) \in \Gamma_m[x] : \textit{there exists } a(x) \in \Gamma_m[x] \textit{ such that } a(x) + pb(x) \in S\}.$$

Note that $|S| \leq |S_0| \cdot |S_1|$. Moreover, since S is a $\mathbb{Z}_{p^2}$-module, we have $S_0 \subseteq S_1$.

It follows from Definition 2.1 that if $S \subseteq \mathrm{GR}(p^2, m)[x]$ is a finite $\mathbb{Z}_{p^2}$-module, then any element $f(x)$ of S is represented as

$$f(x) = a(x) + pb(x)$$

such that $a(x) \in S_0$ and $b(x) \in S_1$ are uniquely determined elements.

Example 2.2. Let $S = \{x + 2x^2, 2x, x + 2(x + x^2), 0\} \subseteq \mathrm{GR}(2^2, m)[x]$. It is easy to observe that S is a $\mathbb{Z}_4$-module. The corresponding subsets S_0 and S_1 are

$$S_0 = \{0, x\} \text{ and } S_1 = \{x, x^2, x^2 + x, 0\}.$$

Now we prove some lemmas that we use later in this section as well as in section 3.

LEMMA 2.3. *Let $S \subseteq \mathrm{GR}(p^2, m)[x]$ be a finite $\mathbb{Z}_{p^2}$-module and $S_1 \subseteq \Gamma_m[x]$ be the subset defined in Definition* 2.1. *Let $T \subseteq \Gamma_m$ be a subset. If the condition*

$$\begin{gathered}\textit{for each } h(x) \in \rho(S_1),\\ h(\nu) = 0 \textit{ for each } \nu \in \rho(T) \Rightarrow h(x) \textit{ is the zero polynomial}\end{gathered} \tag{2.1}$$

holds, then we have

$$\begin{gathered}\textit{for each } f(x) \in S,\\ f(\alpha) = 0 \textit{ for each } \alpha \in T \Rightarrow f(x) \textit{ is the zero polynomial.}\end{gathered}$$

Similarly if the condition

$$\begin{gathered}\textit{for each } h(x) \in \rho(S_1),\\ \mathrm{tr}_m(h(\nu)) = 0 \textit{ for each } \nu \in \rho(T) \Rightarrow h(x) \textit{ is the zero polynomial}\end{gathered} \tag{2.2}$$

holds, then we have

$$\begin{gathered}\textit{for each } f(x) \in S,\\ \mathrm{Tr}_m(f(\alpha)) = 0 \textit{ for each } \alpha \in T \Rightarrow f(x) \textit{ is the zero polynomial.}\end{gathered}$$

Proof. For a given $f(x) \in S$, let $a(x) \in S_0$ and $b(x) \in S_1$ be the elements such that $f(x) = a(x) + pb(x)$. Moreover let $a^{(1)}(x) = \rho(a(x)) \in \rho(S_0) \subseteq \rho(S_1)$ and $b^{(1)}(x) = \rho(b(x)) \in \rho(S_1)$.

Assume first that (2.1) holds and also let $f(x)$ be any element of S such that $f(\alpha) = 0$ for each $\alpha \in T$. Then $\rho(f(\alpha)) = 0$ for each $\alpha \in T$ and hence $a^{(1)}(\nu) = 0$ for each $\nu \in \rho(T)$. By (2.1) we have $a^{(1)}(x) = 0$. Since ρ is one-to-one on $\Gamma_m[x]$, we obtain that $a(x) = 0$. Hence $f(x) = pb(x)$. If $b^{(1)}(\nu) = 0$ for each $\nu \in \rho(T)$, then we have $b(x) = 0$ as above and $f(x) = 0$. Otherwise, if there exists $\nu \in \rho(T)$ such that $b^{(1)}(\nu) \neq 0$, then there exists $\alpha \in T$ such that $b(\alpha) = b_0 + pb_1$ with $b_0 \neq 0$ and hence $f(\alpha) \neq 0$.

Next we assume that (2.2) holds and also we assume that $f(x)$ is an element of S such that $\mathrm{Tr}_m(f(\alpha)) = 0$ for each $\alpha \in T$. Then as above we have $\mathrm{tr}_m(a^{(1)}(\nu)) = 0$ for each $\nu \in \rho(T)$. Using (2.2) we obtain that $a(x) = 0$ and hence $f(x) = pb(x)$. Similarly we also obtain that $b(x) = 0$. □

For any integer j with $1 \leq j \leq p^m - 1$, its p-cyclotomic coset modulo $p^m - 1$ is defined as

$$\begin{aligned}B_j = \{a :\ & 0 \leq a \leq p^m - 2 \text{ and } a \equiv jp^l \mod (p^m - 1)\\ & \text{for some integer } 0 \leq l \leq m - 1\}.\end{aligned}$$

For $1 \leq j \leq p^{m/2}$, let $l = \lfloor m/2 \rfloor$ and $0 \leq j_0, \ldots, j_l \leq p-1$ be the integers such that $j = j_0 + j_1 p + \cdots + j_l p^l$ and hence modulo $(p^m - 1)$ we have

$$(2.3) \qquad \begin{bmatrix} j \\ pj \\ \vdots \\ p^{m-1}j \end{bmatrix} \equiv \begin{bmatrix} j_0 & j_1 & \cdots & j_{l-1} & j_l & 0 & \cdots & 0 \\ 0 & j_0 & \cdots & j_{l-2} & j_{l-1} & j_l & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ j_1 & j_2 & \cdots & j_l & 0 & 0 & \cdots & j_0 \end{bmatrix} \begin{bmatrix} 1 \\ p \\ \vdots \\ p^{l-1} \\ p^l \\ p^{l+1} \\ \vdots \\ p^{m-1} \end{bmatrix}.$$

Note that for j in (2.3), $p \not| j$ means that $j_0 \neq 0$. Using the definition of B_j and the observation in (2.3), the following lemma readily follows.

Lemma 2.4. *Let j be an integer with $1 \leq j \leq p^{m/2}$. Then the cardinality of its p-cyclotomic coset B_j modulo $p^m - 1$ is m. Moreover, if i and j are positive integers with $i < j \leq p^{m/2}$ and $p \not| i$, $p \not| j$, then $B_i \cap B_j = \emptyset$.*

For a nonnegative integer h, let $I(h)$ denote the set of nonnegative integers

$$I(h) = \{i : i \not\equiv 0 \mod p \text{ and } 0 \leq i \leq h\}.$$

Note that $|I(h)| = h - \lfloor \frac{h}{p} \rfloor$.

Lemma 2.5. *Let D be a positive integer with $D \leq p^{m/2}$ and let M be the set of polynomials in $\mathbb{F}_{p^m}[x]$ defined as*

$$M = \left\{ g(x) \in \mathbb{F}_{p^m}[x] : g(x) = \sum_{i \in I(D)} g_i x^i \right\}.$$

If $g(x) \in M$, $a \in \mathbb{F}_p$, and $a + \mathrm{tr}_m(g(\omega^l)) = 0$ for each $1 \leq l \leq p^m - 1$, then $a = 0$ and $g(x) = 0$.

Proof. For $a \in \mathbb{F}_{p^m}$ and $g(x) \in M$ with $a + \mathrm{tr}_m(g(\omega^l)) = 0$ for each $1 \leq l \leq p^m - 1$, let $f(x) \in \mathbb{F}_{p^m}[x]$ such that $\deg f(x) \leq p^m - 2$ and

$$f(x) \equiv a + \mathrm{tr}_m(g(x)) \mod (x^{p^m - 1} - 1).$$

Then $f(\omega^l) = a + \mathrm{tr}_m(g(\omega^l)) = 0$ for each $1 \leq l \leq p^m - 1$. As $\deg f(x) \leq p^m - 2$, we also obtain that $f(x) = 0$. Assume that $g(x)$ is not the zero polynomial, since otherwise the proof is clear. For each monomial $g_i x^i$ in $g(x)$ with a nonzero coefficient g_i, let $f_i(x) \in \mathbb{F}_{p^m}[x]$ such that $\deg f_i(x) \leq p^m - 2$ and

$$(2.4) \qquad f_i(x) \equiv \mathrm{tr}_m(g_i x^i) \mod (x^{p^m - 1} - 1).$$

We have $i \in I(D)$,

$$f_i(x) \equiv g_i x^i + g_i^p x^{ip} + \cdots + g_i^{p^{m-1}} x^{ip^{m-1}} \mod (x^{p^m - 1} - 1)$$

and hence the set of the degrees of the monomials in $f_i(x)$ with nonzero coefficients is the p-cylotomic coset B_i of i modulo $p^m - 1$. As $0 = f(x)$ is the sum of a and the sum of the polynomials in (2.4) as $g_i x^i$ runs through the monomials in $g(x)$ with nonzero coefficients, using Lemma 2.4 we obtain that $a = 0$ and $g(x) = 0$. ∎

Definition 2.6. *For a prime number p we define a weight function w_p on $\mathbb{N}$ as*

$$\begin{aligned} w_p : \mathbb{N} &\rightarrow \mathbb{N} \\ a &\mapsto \textit{the sum of digits of the representation of } a \textit{ in base } p. \end{aligned}$$

In other words, if $a = \sum_{i \geq 0} a_i p^i$ with $0 \leq a_i \leq p-1$ for all $i \geq 0$, then $w_p(a) = \sum_{i \geq 0} a_i$.

We recall that the *weighted degree* (cf. [K-H-C]) D_f of a polynomial $f(x) \in \mathrm{GR}(p^2, m)[x]$ is defined as

$$D_f = \max\{p \deg(a(x)), \deg(b(x))\},$$

where $a(x), b(x) \in \Gamma_m[x]$ are the uniquely determined polynomials such that $f(x) = a(x) + pb(x)$.

Let $f(x) = a(x) + pb(x)$ be a nondegenerate polynomial with $a(x), b(x) \in \Gamma_m[x]$. We recall (see [L-O]) some definitions which depend on $f(x)$. Let $I_f, J_f \subseteq \mathbb{N}$ be subsets defined as

$$a(x) = \sum_{i \in I_f} a_i x^i \text{ and } b(x) = \sum_{j \in J_f} b_j x^j, \text{ where } a_i, b_j \in \Gamma_m \setminus \{0\}.$$

We define nonnegative integers W_f, l_f, and h_f as

$$W_f = \max\{p \ \max\{w_p(i) \mid i \in I_f\}, \ \max\{w_p(j) \mid j \in J_f\}\},$$

$$l_f = \left\lceil \frac{m}{W_f} \right\rceil - 1 \text{ and } h_f = \left\lfloor \frac{m}{W_f} \right\rfloor.$$

The following result is proved in [L-O].

Theorem 2.7. *For a nondegenerate polynomial $f(x) \in \mathrm{GR}(p^2, m)[x]$, we have*

$$\left| \sum_{a \in \mathbb{Z}_{p^2} \setminus p\mathbb{Z}_{p^2}} \sum_{x \in \Gamma_m} e^{2\pi i \frac{\mathrm{Tr}_m(af(x))}{p^2}} \right| \leq p^{l_f+1} \left\lfloor \frac{p^{h_f} \frac{p^2-p}{2} (D_f - 1) \left\lfloor 2p^{\frac{m}{2} - h_f} \right\rfloor}{p^{l_f+1}} \right\rfloor.$$

For a positive integer D, let $S(D) \subseteq \mathrm{GR}(p^2, m)[x]$ be the finite $\mathbb{Z}_{p^2}$-module defined as

$$S(D) = \left\{ f(x) \in \mathrm{GR}(p^2, m)[x] : f(x) = \sum_{i \in I(D)} f_i x^i \text{ and } D_f \leq D \right\}. \tag{2.5}$$

For the subsets $S(D)_0, S(D)_1 \in \Gamma_m[x]$ defined in Definition 2.1 we have

$$S(D)_0 = \left\{ a(x) \in \Gamma_m[x] : a(x) = \sum_{i \in I(\lfloor \frac{D}{p} \rfloor)} a_i x^i \right\},$$

$$S(D)_1 = \left\{ b(x) \in \Gamma_m[x] : b(x) = \sum_{i \in I(D)} a_i x^i \right\},$$

and hence

$$|S(D)| = p^{m\left(D - \lfloor \frac{D}{p^2} \rfloor\right)}. \tag{2.6}$$

For each $u \in \mathbb{Z}_{p^2}$, recall that its homogeneous weight (cf. [C-H], [L-B]) $w_{\mathrm{hom}}(u)$ is defined as

$$w_{\mathrm{hom}}(u) = \begin{cases} 0 & \text{if } u = 0, \\ p & \text{if } u \in p\mathbb{Z}_{p^2} \setminus \{0\}, \\ p-1 & \text{if } u \in \mathbb{Z}_{p^2} \setminus p\mathbb{Z}_{p^2}. \end{cases}$$

Moreover, for each $l \geq 2$ and $u_1, \ldots, u_l \in \mathbb{Z}_{p^2}$, we also have $w_{\mathrm{hom}}(u_1, \ldots, u_l) = \sum_{i=1}^{l} w_{\mathrm{hom}}(u_i)$ by definition.

For $n \geq 1$ we recall that the Gray map (cf. [C], [G-S], [L-B], [L-S]) Φ over $\mathbb{Z}_{p^2}^n$ is defined as follows: For $u \in \mathbb{Z}_{p^2}$ let $u = r_0(u) + pr_1(u)$ with $r_0(u),\ r_1(u) \in \{0, 1, \ldots, p-1\}$. We denote the addition modulo p as $\oplus$. For $(u_0, u_1, \ldots, u_{n-1}) \in \mathbb{Z}_{p^2}^n$, we have $\Phi(u_0, u_1, \ldots, u_{n-1}) = (a_0, a_1, \ldots, a_{pn-1}) \in \mathbb{F}_p^{pn}$ such that for $0 \leq j \leq p-1$ and $0 \leq t \leq n-1$, $a_{jn+t} = r_1(u_t) \oplus jr_0(u_t)$. It follows that for $(u_0, u_1, \ldots, u_{n-1}) \in \mathbb{Z}_{p^2}^n$,

$$w_{\mathrm{hom}}(u_0, u_1, \ldots, u_{n-1}) = w_{\mathrm{H}}\left(\Phi(u_0, u_1, \ldots, u_{n-1})\right),$$

where $w_{\mathrm{H}}(\cdot)$ is the Hamming weight on $\mathbb{F}_p^{pn}$ (cf. [L-B]).

Now we construct a family of p-ary codes generalizing the family of binary codes of [S-K-H]. Note that there is another class of binary codes generalizing the Kerdock and Delsarte–Goethals codes using the ring $\mathbb{Z}_{2^k}$ (cf. [C]).

Definition 2.8. *For $1 \leq D \leq p^{m/2}$, let $C(D)$ be the $\mathbb{Z}_{p^2}$-linear code of length p^m defined as*

$$C(D) = \left\{ \left(\mathrm{Tr}_m(f(0)) + u, \mathrm{Tr}_m(f(\beta)) + u, \ldots, \mathrm{Tr}_m(f(\beta^{p^m-1})) + u \right) \mid f(x) \in S(D) \text{ and } u \in \mathbb{Z}_{p^2} \right\}.$$

The image $\Phi(C(D))$ of $C(D)$ under the Gray map Φ is a p-ary code of length p^{m+1}. From Lemmas 2.3 and 2.4 we obtain that the size of $\Phi(C(D))$ is $p^2|S(D)|$.

Using Theorem 2.7, we may obtain a lower bound for the minimum distance of $\Phi(C(D))$. For $p = 2$, our lower bound coincides with the lower bound of [S-K-H]. We need a further definition in order to state the lower bound on the minimum distance.

Definition 2.9. *For $1 \leq D \leq p^{m/2}$, let W_D, l_D, and h_D be the nonnegative integers defined as*

$$W_D = \max\{W_f \mid f(x) \in S(D) \setminus \{0\}\}, \ \ l_D = \left\lceil \frac{m}{W_D} \right\rceil - 1, \ \text{ and } \ h_D = \left\lfloor \frac{m}{W_D} \right\rfloor.$$

Note that $l_D = \min\{l_f : f(x) \in S(D) \setminus \{0\}\}$ and $h_D = \min\{h_f : f(x) \in S(D) \setminus \{0\}\}$.

The following theorem follows from Theorem 2.7 and (2.6).

Theorem 2.10. *For $1 \leq D \leq p^{m/2}$, $\Phi(C(D))$ is a p-ary code of length p^{m+1} of minimum distance*

$$d_{\min} \geq p^{m+1} - p^m - p^{l_D} \left\lfloor \frac{p^{h_D \frac{p^2-p}{2}} (D-1) \left\lfloor 2p^{\frac{m}{2} - h_D} \right\rfloor}{p^{l_D+1}} \right\rfloor \tag{2.7}$$

and of size

$$\left|\Phi(C(D))\right| = p^2 \cdot p^{m\left(D - \lfloor \frac{D}{p^2} \rfloor\right)}. \tag{2.8}$$

3. p-ary sequences with low correlation. In this section we obtain several families of pairwise cyclically distinct p-ary sequences with low correlation.

We begin with a simple lemma that we use later.

LEMMA 3.1. *For $a \in \mathrm{GR}(p^2, m) \setminus \{0\}$ and a nonnegative integer i, we have*

$$a(1-\beta^i) = 0 \implies \beta^i = 1.$$

Proof. Let $a = a_0 + pa_1$ with $a_0, a_1 \in \Gamma_m$. If $a_0 \neq 0$, then a is a unit and we get the conclusion. If $a_0 = 0$, then $a = pa_1$ and hence $1-\beta^i$ belongs to the maximal ideal (p) of $\mathrm{GR}(p^2, m)$. Therefore $1 - \omega^i = 0$, which implies that $\beta^i = 1$. □

For $f(x) \in S(D)$ and for each $i \geq 0$, we have $\mathrm{Tr}_m(f(\beta^i)) = \mathrm{Tr}_m(f(\beta^{i+(p^m-1)}))$. Therefore the period of $\mathbb{Z}_{p^2}$-sequence $\{\mathrm{Tr}_m(f(\beta^i))\}_{i=0}^{\infty}$ divides $p^m - 1$. We first study the exact periods of the sequences in greater detail.

LEMMA 3.2. *For a positive divisor t of $p^m - 1$ and $f(x) = \sum_{s\in I(D)} f_s x^s \in S(D) \setminus \{0\}$, the period of $\{\mathrm{Tr}_m(f(\beta^i))\}_{i=0}^{\infty}$ is t if and only if $\gcd(\gcd\{s \in I(D) : f_s \neq 0\}, p^m - 1) = \frac{p^m-1}{t}$.*

Proof. Let $u = \gcd(\gcd\{s \in I(D) : f_s \neq 0\}, p^m - 1)$ and $t_1 = \frac{p^m-1}{u}$. Assume that the period of $\{\mathrm{Tr}_m(f(\beta^i))\}_{i=0}^{\infty}$ is t. For each $i \geq 0$, we have $\mathrm{Tr}_m(f(\beta^i)) = \mathrm{Tr}_m(f(\beta^{i+t}))$. Then

$$\mathrm{Tr}_m\left(\sum_{s\in I(D)} f_s(1-\beta^{ts})\beta^{is}\right) = 0$$

for each $i \geq 0$. Since $\sum_{s\in I(D)} f_s(1-\beta^{ts})x^s \in S(D)$, using Lemmas 2.3 and 2.5 we obtain that $f_s(1-\beta^{ts}) = 0$ for each $s \in I(D)$. By Lemma 3.1, we have $\beta^{ts} = 1$ for each $s \in I(D)$ with $f_s \neq 0$. Then $(p^m-1)|(ts)$ for each $s \in I(D)$ with $f_s \neq 0$ and hence $\frac{p^m-1}{t}|u$, i.e., t_1 divides t. On the other hand, note that $\beta^{st_1} = \beta^{\frac{s}{u}(p^m-1)} = 1$ for each $s \in I(D)$ with $f_s \neq 0$. Hence $\mathrm{Tr}_m(f(\beta^{i+t_1})) = \mathrm{Tr}_m(f(\beta^i))$ for each $i \geq 0$, so t divides t_1.

Conversely assume that t is a positive divisor of p^m-1 and $f(x) = \sum_{s\in I(D)} f_s x^s \in S(D) \setminus \{0\}$ such that $u = \frac{p^m-1}{t}$. Then $\beta^{st} = \beta^{\frac{s}{u}(p^m-1)} = 1$ for each $s \in I(D)$ with $f_s \neq 0$ and hence $\mathrm{Tr}_m(f(\beta^{i+t})) = \mathrm{Tr}_m(f(\beta^i))$ for each $i \geq 0$. Let $t_2 \leq t$ be the period of $\{\mathrm{Tr}_m(f(\beta^i))\}_{i=0}^{\infty}$. If $t_2 < t$, then by the first part of the proof above, we have $u = \frac{p^m-1}{t_2} \neq \frac{p^m-1}{t}$, which is a contradiction. This completes the proof. □

The following lemma is used in the proof of Proposition 3.5.

LEMMA 3.3. *For each positive divisor t of (p^m-1), the mapping $g(x) \in S(\lfloor \frac{D}{t} \rfloor) \mapsto f(x) = g(x^t) \in S(D)$ gives a one-to-one correspondence between $S(\lfloor \frac{D}{t} \rfloor)$ and the polynomials $f(x) \in S(D)$ such that $\{\mathrm{Tr}_m(f(\beta^i))\}_{i=0}^{\infty}$ is a sequence of period dividing $\frac{p^m-1}{t}$.*

Proof. For $g(x) \in S(\lfloor \frac{D}{t} \rfloor) \setminus \{0\}$ and $f(x) = g(x^t) = \sum_{s\in I(D)} f_s x^s$, we have $f(x) \in S(D) \setminus \{0\}$ and $t|\gcd(\gcd\{s \in I(D) : f_s \neq 0\}, p^m - 1)$. Then the period of $\{\mathrm{Tr}_m(f(\beta^i))\}_{i=0}^{\infty}$ is a divisor of $\frac{p^m-1}{t}$ by Lemma 3.2. Conversely if $f(x) = \sum_{s\in I(D)} f_s x^s \in S(D) \setminus \{0\}$ with the period t_1 of $\{\mathrm{Tr}_m(f(\beta^i))\}_{i=0}^{\infty}$ such that $t_1|\frac{p^m-1}{t}$, then $\gcd(\gcd\{s \in I(D) : f_s \neq 0\}, p^m - 1) = \frac{p^m-1}{t_1} = ta$ for a positive integer a. Hence $t|s$ for each $s \in I(D)$ with $f_s \neq 0$ and there exists a uniquely determined $g(x) \in S(\lfloor \frac{D}{t} \rfloor)$ such that $g(x^t) = f(x)$. □

Remark 3.4. Using the similar mapping in Lemma 3.3 for $pS(D)_1$, for each positive divisor t of (p^m-1), we also obtain a one-to-one correspondence between $S(\lfloor \frac{D}{t} \rfloor)_1$ and the polynomials $f(x) \in pS(D)_1$ such that $\{\mathrm{Tr}_m(f(\beta^i))\}_{i=0}^{\infty}$ has period dividing $\frac{p^m-1}{t}$.

In the next proposition, for each positive divisor t of p^m-1, we compute the number of polynomials $f(x)$ in $S(D)$ such that the $\mathbb{Z}_{p^2}$-sequence $\{\mathrm{Tr}_m(f(\beta^i))\}_{i=0}^{\infty}$ has period $\frac{p^m-1}{t}$.

Proposition 3.5. *For each positive divisor t of (p^m-1), we have*

$$\left|\left\{f(x) \in S(D) : \{\mathrm{Tr}_m(f(\beta^i))\}_{i=0}^{\infty} \textit{ has period } \frac{p^m-1}{t}\right\}\right|$$

$$= \sum_{l \mid \frac{p^m-1}{t}} \mu(l) \left| S\left(\left\lfloor \frac{D}{l \cdot t} \right\rfloor\right)\right|,$$

where $\mu(\cdot)$ is the Möbius function.

Proof. For positive integers u and v, let

$$h(u) = |\{f(x) \in S(D) : \text{ the period of } \{\mathrm{Tr}_m(f(\beta^i))\}_{i=0}^{\infty} \text{ is } u\}|$$

and

$$H(v) = |\{f(x) \in S(D) : \text{ the period of } \{\mathrm{Tr}_m(f(\beta^i))\}_{i=0}^{\infty} \text{ is a positive divisor of } v\}|.$$

Then $h(u) = 0$ if $u \geq 1$ such that u does not divide (p^m-1). Moreover, for each positive integer v,

$$H(v) = \sum_{u \mid v} h(u).$$

Using the Möbius inversion formula (cf. [L-N, Theorem 3.24]), for each positive integer u we obtain that

$$h(u) = \sum_{v \mid u} \mu(v) H\left(\frac{u}{v}\right).$$

In particular, for $u = \frac{p^m-1}{t}$ with a positive divisor t of (p^m-1),

$$h\left(\frac{p^m-1}{t}\right) = \sum_{l \mid \frac{p^m-1}{t}} \mu(l) H\left(\frac{p^m-1}{l \cdot t}\right).$$

Using Lemma 3.3, we obtain $H(\frac{p^m-1}{l \cdot t}) = |S(\lfloor \frac{D}{l \cdot t} \rfloor)|$, which completes the proof. □

Using the same method as the proof of Proposition 3.5 and Remark 3.4, we also obtain the following result.

Proposition 3.6. *For each positive divisor t of (p^m-1), we have*

$$\left|\left\{f(x) \in pS(D)_1 : \{\mathrm{Tr}_m(f(\beta^i))\}_{i=0}^{\infty} \textit{ has period } \frac{p^m-1}{t}\right\}\right|$$

$$= \sum_{l \mid \frac{p^m-1}{t}} \mu(l) \left| S\left(\left\lfloor \frac{D}{l \cdot t} \right\rfloor\right)_1\right|,$$

where $\mu(\cdot)$ is the Möbius function.

For $n = p^m - 1$ we recall that the generalized Nechaev–Gray map (cf. [N], [L-B], [L-S]) Ψ over $\mathbb{Z}_{p^2}^n$ is defined as follows: for $u \in \mathbb{Z}_{p^2}$ let $u = r_0(u) + pr_1(u)$ with $r_0(u)$, $r_1(u) \in \{0, 1, \ldots, p-1\}$. Let $\oplus$ denote the addition modulo p. For $(u_0, u_1, \ldots, u_{n-1}) \in \mathbb{Z}_{p^2}^n$, we have $\Psi(u_0, u_1, \ldots, u_{n-1}) = (a_0, a_1, \ldots, a_{pn-1}) \in \mathbb{F}_p^{pn}$ such that for $0 \le j \le p-1$ and $0 \le t \le n-1$, $a_{jn+t} = r_1((1-p)^t u_t) \oplus jr_0((1-p)^t u_t)$. It is known that if C is a cyclic code of length $p^m - 1$ over $\mathbb{Z}_{p^2}$, then $\Psi(C)$ is a cyclic code of length $p(p^m-1)$ over $\mathbb{F}_p$ (cf. [L-B, Corollary 2.5]). Therefore the generalized Nechaev–Gray map may be used for constructing sequences.

PROPOSITION 3.7. *Assume that $u \in \mathbb{Z}_{p^2}$. For $f(x) \in S(D)$ such that the corresponding $\mathbb{Z}_{p^2}$-sequence $\{\mathrm{Tr}_m(f(\beta^i))\}_{i=0}^{\infty}$ has period $p^m - 1$, we have the following:*

(i) *if $\rho(f(x)) \neq 0$, then the p-ary sequence $\{\Psi(\mathrm{Tr}_m(f(\beta^i)) + u)\}_{i=0}^{\infty}$ has period $p(p^m-1)$;*

(ii) *if $\rho(f(x)) = 0$ and $\rho(u) \neq 0$, then the p-ary sequence $\{\Psi(\mathrm{Tr}_m(f(\beta^i)) + u)\}_{i=0}^{\infty}$ has period $p(p^m-1)$;*

(iii) *if $\rho(f(x)) = 0$ and $\rho(u) = 0$, then the p-ary sequence $\{\Psi(\mathrm{Tr}_m(f(\beta^i)) + u)\}_{i=0}^{\infty}$ has period $p^m - 1$.*

Proof. Every $z \in \mathbb{Z}_{p^2}$ can be written uniquely as $z = r_0(z) + pr_1(z)$ with $r_0(z), r_1(z) \in \{0, 1, \ldots, p-1\}$. Let $\oplus$ and $\ominus$ denote the addition and subtraction, respectively, modulo p while $+$ and $-$ denote the addition and subtraction in $\mathbb{Z}_{p^2}$.

For $f(x) \in S(D)$ satisfying the condition of the proposition and $u \in \mathbb{Z}_{p^2}$, we first observe that the period of the $\mathbb{Z}_{p^2}$-sequence $\{\mathrm{Tr}_m(f(\beta^i)) + u\}_{i=0}^{\infty}$ is also $p^m - 1$.

By the definition of Ψ, for $0 \le t \le (p^m - 1) - 1$ and $j \ge 0$, the $(j(p^m-1)+t)$th term of the p-ary sequence $\{\Psi(\mathrm{Tr}_m(f(\beta^i)) + u)\}_{i=0}^{\infty}$ is given by

$$\begin{aligned}
&r_1\left((1-p)^t(\mathrm{Tr}_m(f(\beta^t)) + u)\right) \oplus jr_0\left((1-p)^t(\mathrm{Tr}_m(f(\beta^t)) + u)\right)\\
&= r_1\left((1+jp)(1-p)^t(\mathrm{Tr}_m(f(\beta^t)) + u)\right)\\
&= r_1\left((1-p)^{j(p^m-1)+t}(\mathrm{Tr}_m(f(\beta^{j(p^m-1)+t})) + u)\right).
\end{aligned}$$

The period of $\{\mathrm{Tr}_m(f(\beta^i)) + u\}_{i=0}^{\infty}$ is $p^m - 1$, while the period of $\{(1-p)^i\}_{i=0}^{\infty}$ is p. Hence, the period T of $\{\Psi(\mathrm{Tr}_m(f(\beta^i)) + u)\}_{i=0}^{\infty} = \left\{r_1\left((1-p)^i(\mathrm{Tr}_m(f(\beta^i)) + u)\right)\right\}_{i=0}^{\infty}$ divides $p(p^m-1)$.

As the period of $\{\Psi(\mathrm{Tr}_m(f(\beta^i)) + u)\}_{i=0}^{\infty}$ is T, for each $i \ge 0$ and $j \ge 0$ we have

$$\begin{aligned}
&r_1\left((1-p)^{i+j(p^m-1)}(\mathrm{Tr}_m(f(\beta^{i+j(p^m-1)})) + u)\right)\\
&= r_1\left((1-p)^{i+j(p^m-1)+T}(\mathrm{Tr}_m(f(\beta^{i+j(p^m-1)+T})) + u\right)
\end{aligned}$$

and hence

$$\begin{aligned}
&r_1\left(\mathrm{Tr}_m(f(\beta^i)) + u\right) \ominus (i-j)r_0\left(\mathrm{Tr}_m(f(\beta^i)) + u\right)\\
&= r_1\left(\mathrm{Tr}_m(f(\beta^{i+T})) + u\right) \ominus (i+T-j)r_0\left(\mathrm{Tr}_m(f(\beta^{i+T})) + u\right).
\end{aligned} \tag{3.1}$$

Comparing (3.1) for different values of j, we obtain

$$r_0\left(\mathrm{Tr}_m(f(\beta^i)) + u\right) = r_0\left(\mathrm{Tr}_m(f(\beta^{i+T})) + u\right) \tag{3.2}$$

for each $i \ge 0$. Using (3.1) with $i = j$, we also get

$$\begin{aligned}
r_1\left(\mathrm{Tr}_m(f(\beta^i)) + u\right) = {}& r_1\left(\mathrm{Tr}_m(f(\beta^{i+T})) + u\right)\\
&\ominus Tr_0\left(\mathrm{Tr}_m(f(\beta^{i+T})) + u\right).
\end{aligned} \tag{3.3}$$

It follows from (3.2) and (3.3) that

$$\begin{aligned}
r_0\left(\mathrm{Tr}_m(f(\beta^i)) + u\right) &= r_0\left(\mathrm{Tr}_m(f(\beta^{i+pT})) + u\right),\\
r_1\left(\mathrm{Tr}_m(f(\beta^i)) + u\right) &= r_1\left(\mathrm{Tr}_m(f(\beta^{i+pT})) + u\right),
\end{aligned}$$

and hence

$$\mathrm{Tr}_m(f(\beta^i)) + u = \mathrm{Tr}_m(f(\beta^{i+pT})) + u$$

for each $i \geq 0$. Therefore $(p^m - 1)|(pT)$, which implies that $(p^m - 1)|T$. Moreover, recall also that T divides $p(p^m - 1)$, so

$$\text{either } T = p^m - 1 \text{ or } T = p(p^m - 1). \tag{3.4}$$

Now we prove that $T = p(p^m - 1)$ for (i) and (ii). Using (3.4) we assume the contrary that $T = p^m - 1$. As $\beta^T = 1$, by (3.3) we obtain that

$$r_0(\mathrm{Tr}_m(f(\beta^i)) + u) = \mathrm{tr}_m(\rho(f)(\omega^i)) + \rho(u) = 0$$

for each $i \geq 0$. Then by Lemma 2.5 we have $\rho(u) = 0$ and $\rho(f(x)) = 0$, which completes the proof for (i) and (ii).

Next we assume $\rho(f(x)) = 0$, $\rho(u) = 0$ and consider the remaining case. For each $i \geq 0$, the ith and $(i + (p^m - 1))$th terms of $\{\Psi(\mathrm{Tr}_m(f(\beta^i)) + u)\}_{i=0}^{\infty}$ are

$$r_1\left(\mathrm{Tr}_m(f(\beta^i)) + u\right) \ominus i r_0\left(\mathrm{Tr}_m(f(\beta^i)) + u\right)$$

and

$$r_1\left(\mathrm{Tr}_m(f(\beta^i)) + u\right) \ominus (i-1) r_0\left(\mathrm{Tr}_m(f(\beta^i)) + u\right), \tag{3.5}$$

respectively. Therefore, as $\rho(f(x)) = 0$, the ith and $(i + (p^m - 1))$th terms of $\{\Psi(\mathrm{Tr}_m(f(\beta^i)) + u)\}_{i=0}^{\infty}$ are equal for each $i \geq 0$ if and only if

$$r_0(u) = \rho(u) = 0. \tag{3.6}$$

We complete the proof for (iii) using (3.4) and (3.6). □

Now we begin our construction of families of pairwise cyclically distinct p-ary sequences of low correlation.

For a p-ary sequence $\{s(i)\}_{i=0}^{\infty}$ and $\tau \geq 0$, the *cyclic shift of* $\{s(i)\}_{i=0}^{\infty}$ *by* τ is the p-ary sequence $\{s(i+\tau)\}_{i=0}^{\infty}$. For two p-ary sequences $\{s_1(i)\}_{i=0}^{\infty}$ and $\{s_2(i)\}_{i=0}^{\infty}$, we say $\{s_1(i)\}_{i=0}^{\infty}$ and $\{s_2(i)\}_{i=0}^{\infty}$ are *cyclically distinct* if for each $\tau \geq 1$ neither is $\{s_1(i)\}_{i=0}^{\infty}$ the cyclic shift of $\{s_2(i)\}_{i=0}^{\infty}$ by τ nor is $\{s_2(i)\}_{i=0}^{\infty}$ the cyclic shift of $\{s_1(i)\}_{i=0}^{\infty}$ by τ.

Let $\mathcal{P}_D^1$ be the subset of $S(D) \times \mathbb{Z}_{p^2}$ defined as

$$\mathcal{P}_D^1 = \Big\{(f(x), u) \in S(D) \times \mathbb{Z}_{p^2} : \quad \rho(f(x)) \neq 0, \\ \text{and } \{\mathrm{Tr}_m(f(\beta^i))\}_{i=0}^{\infty} \text{ has period } p^m - 1\Big\}.$$

Using Propositions 3.5 and 3.6, we obtain that

$$|\mathcal{P}_D^1| = p^2 \left(\sum_{l|(p^m-1)} \mu(l) \left\{ p^{m\left(\lfloor \frac{D}{l} \rfloor - \lfloor \frac{D}{p^2 l} \rfloor\right)} - p^{m\left(\lfloor \frac{D}{l} \rfloor - \lfloor \frac{D}{pl} \rfloor\right)} \right\} \right). \tag{3.7}$$

By Proposition 3.7, for each $(f(x), u) \in \mathcal{P}_D^1$, the corresponding p-ary sequence $\{\Psi(\mathrm{Tr}_m(f(\beta^i))+u)\}_{i=0}^{\infty}$ has period $p(p^m-1)$. For $(f(x), u) \in \mathcal{P}_D^1$, $0 \leq t \leq (p^m-1)-1$, and $0 \leq j \leq p-1$, let $g(x) = (1+p)^j(1-p)^t f(\beta^t x)$ and $v = (1+p)^j(1-p)^t u$. Note that $(g(x), v) \in \mathcal{P}_D^1$. Now we prove that, for $0 \leq t \leq (p^m - 1) - 1$ and $0 \leq j \leq$

$p-1$, $(f(x),u)=(g(x),v)$ as elements of $\mathcal{P}_D^1$ if and only if $j=t=0$. Note that $(f(x),u)=(g(x),v)$ as elements of $\mathcal{P}_D^1$ if and only if the corresponding p-ary sequences $\{\Psi(\mathrm{Tr}_m(f(\beta^i))+u)\}_{i=0}^{\infty}$ and $\{\Psi(\mathrm{Tr}_m(g(\beta^i))+v)\}_{i=0}^{\infty}$ are equal. Let $\{s_1(i)\}_{i=0}^{\infty}$ be the p-ary sequence $\{\Psi(\mathrm{Tr}_m(f(\beta^i))+u)\}_{i=0}^{\infty}$ and $\{s_2(i)\}_{i=0}^{\infty}$ be the p-ary sequence $\{\Psi(\mathrm{Tr}_m(g(\beta^i))+v)\}_{i=0}^{\infty}$. Therefore $\{s_2(i)\}_{i=0}^{\infty}$ is the cyclic shift of $\{s_1(i)\}_{i=0}^{\infty}$ by $t+j(p^m-1)$ since both sequences have period $p(p^m-1)$ and, under the notation of the proof of Proposition 3.7, for each $i\geq 0$

$$\begin{aligned} s_2(i) &= r_1\left(\mathrm{Tr}_m(f(\beta^{i+t}))+u\right)\ominus(i+t-j)r_0\left(\mathrm{Tr}_m(f(\beta^{i+t}))+u\right)\\ &= s_1(i+t+j(p^m-1)). \end{aligned}$$

For $(f(x),u),(g(x),v)\in\mathcal{P}_D^1$, we say $(f(x),u)$ and $(g(x),v)$ are *cyclically related* if there exist $0\leq j\leq p-1$ and $0\leq t\leq(p^m-1)-1$ such that $g(x)=(1+p)^j(1-p)^t f(\beta^t x)$ and $v=(1+p)^j(1-p)^t u$. From the arguments above, we observe that cyclically related elements of $\mathcal{P}_D^1$ form an equivalence relation on $\mathcal{P}_D^1$ and each equivalence class has $p(p^m-1)$ elements.

Let $\overline{\mathcal{P}}_D^1$ be the set of these equivalence classes in $\mathcal{P}_D^1$. Then

$$|\overline{\mathcal{P}}_D^1|=\frac{1}{p(p^m-1)}|\mathcal{P}_D^1|. \tag{3.8}$$

Let $\widetilde{\mathcal{P}}_D^1$ be a full set of representatives of the equivalence classes in $\overline{\mathcal{P}}_D^1$.

The element, i.e., the equivalence class, of $\overline{\mathcal{P}}_D^1$ containing $(f(x),u)$ is denoted by $\overline{(f(x),u)}$. Now we prove the following property of the equivalence relation on $\mathcal{P}_D^1$:

$$\begin{aligned} f(x)\in\mathcal{P}_D^1,\quad & u\in\mathbb{Z}_{p^2},\ 0\leq j_1<j_2\leq p-1\\ & \Rightarrow\overline{(f(x),u+j_1p)}\neq\overline{(f(x),u+j_2p)}. \end{aligned} \tag{3.9}$$

Assume the contrary and let $(f(x),u+j_2p)\in\overline{(f(x),u+j_1p)}$. By definition of the equivalence, there exist integers $0\leq j\leq p-1$ and $0\leq t\leq(p^m-1)-1$ such that

$$f(x)=(1+(j-t)p)\,f(\beta^t x) \tag{3.10}$$

and

$$u+j_2p=(1+(j-t)p)\,(u+j_1p). \tag{3.11}$$

As $\rho(f(x))\neq 0$, there exists a coefficient f_s of x^s in $f(x)$ with $\rho(f_s)\neq 0$. From (3.10) we obtain that

$$f_s=(1+(j-t)p)\,\beta^{ts}f_s. \tag{3.12}$$

Let $f_s=a_0+pa_1$ with $a_0,\ a_1\in\Gamma_m$. Then using (3.12) we get

$$\rho(a_0)=\rho(a_0)\rho(\beta^{ts})\ \text{ and }\ \rho(a_1)=\rho\left((j-t)a_0+a_1\right)\rho(\beta^{ts}).$$

As $\rho(a_0)\neq 0$ we have $\rho(\beta^{ts})=1$ and

$$\rho(a_1)=\rho((j-t)a_0+a_1). \tag{3.13}$$

Using $\rho(a_0)\neq 0$ and (3.13) we obtain that $(j-t)\equiv 0 \mod p$. Therefore from (3.11) we get a contradiction.

Now we introduce a new relation on $\mathcal{P}_D^1$: we say that $(f(x), u)$ and $(g(x), v)$ are related in the new sense if there exist $0 \leq j,\ k \leq p-1$, and $0 \leq t \leq (p^m-1)-1$ such that

$$\begin{aligned} g(x) &= (1+p)^j(1-p)^t f(\beta^t x), \\ v &= (1+p)^j(1-p)^t u + kp. \end{aligned} \tag{3.14}$$

It is easy to observe that (3.14) also gives an equivalence relation on $\mathcal{P}_D^1$, and the equivalence relation obtained by the cyclically related elements is finer than the one obtained by (3.14). Let $\widehat{\mathcal{P}}_D^1$ be a full set of representatives of the new equivalence relation. We have $|\widehat{\mathcal{P}}_D^1| = \frac{|\widetilde{\mathcal{P}}_D^1|}{p}$. Moreover we assume, without loss of generality, that the elements of $\widehat{\mathcal{P}}_D^1$ are of the form $(f(x), u)$ with $u \in \{0, 1, \ldots, p-1\} \subseteq \mathbb{Z}_{p^2}$ and $\widehat{\mathcal{P}}_D^1 \subseteq \widetilde{\mathcal{P}}_D^1$.

Let $\mathcal{F}_D^1 \subseteq \mathcal{C}_D^1$ be the chain of families of p-ary sequences defined as

$$\mathcal{F}_D^1 = \big\{\{\Psi(\mathrm{Tr}_m(f(\beta^i)) + u)\}_{i=0}^{\infty} : (f(x), u) \in \widehat{\mathcal{P}}_D^1\big\}$$

and

$$\mathcal{C}_D^1 = \big\{\{\Psi(\mathrm{Tr}_m(f(\beta^i)) + u)\}_{i=0}^{\infty} : (f(x), u) \in \widetilde{\mathcal{P}}_D^1\big\}.$$

THEOREM 3.8. *The families $\mathcal{F}_D^1$ and $\mathcal{C}_D^1$ have the following properties:*

(i) *The period of each sequence in $\mathcal{C}_D^1$ (and hence in $\mathcal{F}_D^1$) is $p(p^m-1)$.*

(ii) *The sequences in $\mathcal{C}_D^1$ (and hence in $\mathcal{F}_D^1$) are pairwise cyclically distinct.*

(iii) $|\mathcal{F}_D^1| = \frac{1}{p^m-1}\sum_{l|(p^m-1)} \mu(l)\{p^{m(\lfloor D/l\rfloor - \lfloor D/p^2 l\rfloor)} - p^{m(\lfloor D/l\rfloor - \lfloor D/pl\rfloor)}\}$ *and* $|\mathcal{C}_D^1| = p|\mathcal{F}_D^1|$, *where $\mu(\cdot)$ is the Möbius function.*

(iv) *For the maximal nontrivial correlation $\theta_{\max}$ of $\mathcal{F}_D^1$, we have*

$$\theta_{\max} \leq \frac{1}{p-1} p^{l_D+1} \left\lfloor \frac{p^{h_D \frac{p^2-p}{2}}(D-1)\left\lfloor 2p^{\frac{m}{2}-h_D}\right\rfloor}{p^{l_D+1}} \right\rfloor + p, \tag{3.15}$$

where l_D and h_D are as in Definition 2.9.

Proof. As $\widetilde{\mathcal{P}}_D^1 \subseteq \mathcal{P}_D^1$, by Proposition 3.7, each sequence in $\mathcal{C}_D^1$ has period $p(p^m-1)$.

Now we prove items (ii) and (iii) together. Let $(f(x), u) \in \widetilde{\mathcal{P}}_D^1$ and $\{s(i)\}_{i=0}^{\infty}$ be the p-ary sequence $\{\Psi(\mathrm{Tr}_m(f(\beta^i)) + u)\}_{i=0}^{\infty}$. Assume that $0 \leq j \leq p-1$ and $0 \leq t \leq (p^m-1)-1$ are integers such that the p-ary sequence $\{s(i+j(p^m-1)+t)\}_{i=0}^{\infty}$ is in $\mathcal{C}_D^1$. Let $(g(x), u_2) \in \widetilde{\mathcal{P}}_D^1$ such that the p-ary sequence $\{\Psi(\mathrm{Tr}_m(g(\beta^i)) + u_2)\}_{i=0}^{\infty}$ is $\{s(i+j(p^m-1)+t)\}_{i=0}^{\infty}$. Let $h(x) = (1+p)^j(1-p)^t f(\beta^t x)$ and $u_1 = u(1+p)^j(1-p)^t \in \mathbb{Z}_{p^2}$. Note that $\overline{(h(x), u_1)} = \overline{(f(x), u)}$ and hence $(h(x), u_1) \notin \widetilde{\mathcal{P}}_D^1$ if either $j \neq 0$ or $t \neq 0$. Let $\{s_1(i)\}_{i=0}^{\infty}$ be the p-ary sequence $\{\Psi(\mathrm{Tr}_m(h(\beta^i)) + u_1)\}_{i=0}^{\infty}$ and $\{s_2(i)\}_{i=0}^{\infty}$ be the p-ary sequence $\{\Psi(\mathrm{Tr}_m(g(\beta^i)) + u_2)\}_{i=0}^{\infty}$. From the proof of Proposition 3.7 we observe that $s_1(i) = s_2(i)$ for each $i \geq 0$. Then for each $i \geq 0$ we have

$$\mathrm{Tr}_m(h(\beta^i)) + u_1 = \mathrm{Tr}_m(g(\beta^i)) + u_2 \tag{3.16}$$

and hence

$$\rho(u_1) + \mathrm{tr}_m(\rho(h)(\omega^i)) = \rho(u_2) + \mathrm{tr}_m(\rho(g)(\omega^i)).$$

As $h(x),\ g(x) \in S(D)$, using Lemmas 2.5 and 2.3 we obtain that $h(x) = g(x)$ and $\rho(u_1) = \rho(u_2)$. Also using (3.16) we obtain that $u_1 = u_2$ and hence $(h(x), u_1) =$

$(g(x), u_2)$. This completes the proof of item (ii). We complete the proof of item (iii) using item (i) and Propositions 3.5 and 3.6.

It remains to prove item (iv). Assume that $(f_1(x), u_1)$, $(f_2(x), u_2) \in \widehat{\mathcal{P}}_D^1$, and let the corresponding p-ary sequences be $\{s_1(i)\}_{i=0}^{\infty} = \{\Psi(\mathrm{Tr}_m(f_1(\beta^i)) + u_1)\}_{i=0}^{\infty}$ and $\{s_2(i)\}_{i=0}^{\infty} = \{\Psi(\mathrm{Tr}_m(f_2(\beta^i)) + u_2)\}_{i=0}^{\infty}$. We consider two cases separately.

Case 1. Correlation at $0 < \tau \leq p(p^m - 1) - 1$. Let $\tau = t + j(p^m - 1)$, where $0 \leq t \leq (p^m - 1) - 1$ and $0 \leq j \leq p - 1$. Let $f_3(x) = (1 + p)^j (1 - p)^t f_1(\beta^t x)$, $u_3 = (1 + p)^j (1 - p)^t u_1$, $f(x) = f_3(x) - f_2(x) \in S(D)$, and $u = u_3 - u_2$.

Assume first that $f(x) \neq 0$. Let

$$\boldsymbol{y} = \Psi\big(\mathrm{Tr}_m(f(\beta^0)) + u, \mathrm{Tr}_m(f(\beta^1)) + u, \ldots, \mathrm{Tr}_m(f(\beta^{(p^m-1)-1})) + u\big). \tag{3.17}$$

Let $\theta_1(\cdot)$ be the exponential sum function on $\mathbb{F}_p^{p(p^m-1)}$ defined in [L-S, section 3]. The correlation between $\{s_1(i)\}_{i=0}^{\infty}$ and $\{s_2(i)\}_{i=0}^{\infty}$ at shift $(t + j(p^m - 1))$ is given by (cf. [L-S, Theorem 3.1])

$$\theta_1(\boldsymbol{y}) = \sum_{l=0}^{p-1} \sum_{x \in \Gamma_m \setminus \{0\}} e^{2\pi i \frac{(1+lp)(\mathrm{Tr}_m(f(x))+u)}{p^2}}. \tag{3.18}$$

Next we assume that $f(x) = 0$. Let $u = r_0(u) + pr_1(u)$ with $r_0(u)$, $r_1(u) \in \{0, 1, \ldots, p-1\}$. Using (3.9) we obtain that $r_0(u) \neq 0$. Then the correlation between $\{s_1(i)\}_{i=0}^{\infty}$ and $\{s_2(i)\}_{i=0}^{\infty}$ at shift $(t + j(p^m - 1))$ is

$$\sum_{l=0}^{p-1} \sum_{x \in \Gamma_m \setminus \{0\}} e^{2\pi i \frac{(1+lp)u}{p^2}} = (p^m - 1) e^{2\pi i \frac{r_0(u)}{p^2}} \sum_{l=0}^{p-1} e^{2\pi i \frac{l r_0(u) + r_1(u)}{p}} = 0.$$

Case 2. Correlation at $\tau = 0$. In this case we have $(f_1(x), u_1) \neq (f_2(x), u_2)$. Let $f(x) = f_1(x) - f_2(x) \in S(D)$ and $u = u_1 - u_2$. Assume first that $f_1(x) \neq f_2(x)$. The correlation in this subcase is also given by the same formula in (3.18).

Next we assume that $f_1(x) = f_2(x)$. Then $\rho(u) \neq 0$ and we obtain that the correlation is 0 as in Case 1.

Therefore in order to complete the proof of item (iv), it is enough to prove that for each $f(x) \in S(D) \setminus \{0\}$ and $u \in \mathbb{Z}_{p^2}$, the absolute value of $\theta_1(\boldsymbol{y})$ given in (3.18) is bounded from above by the value on the right-hand side of (3.15).

Let $f(x) \in S(D) \setminus \{0\}$ and $u \in \mathbb{Z}_{p^2}$. For $0 \leq c \leq (p - 1)$, let

$$\phi_c(f, u) = \sum_{l=0}^{p-1} \sum_{x \in \Gamma_m} e^{2\pi i \frac{(c+lp)(\mathrm{Tr}_m(f(x))+u)}{p^2}}. \tag{3.19}$$

For $1 \leq c \leq (p - 1)$ and $0 \leq l \leq p - 1$, let $0 \leq l_{c^{-1}} \leq p - 1$ be the integer such that $c l_{c^{-1}} \equiv l \mod p$. Then

$$(c + lp)(\mathrm{Tr}_m(f(x)) + u) = (c + c l_{c^{-1}} p)(\mathrm{Tr}_m(f(x)) + u) \tag{3.20}$$

for $1 \leq c \leq (p - 1)$ and $x \in \Gamma_m$. For $1 \leq c \leq (p - 1)$, as c is both invertible in $\mathbb{F}_p$ and in Γ_m, using (3.19) and (3.20) we obtain

$$\phi_c(f, u) = \phi_1(f, u). \tag{3.21}$$

By (3.19) and (3.21), we have

$$\sum_{a\in\mathbb{Z}_{p^2}}\sum_{x\in\Gamma_m} e^{2\pi i\frac{a(\mathrm{Tr}_m(f(x))+u)}{p^2}} = \sum_{c=0}^{p-1}\phi_c(f,u) = \phi_0(f,u) + (p-1)\phi_1(f,u).$$

Hence

$$(p-1)\phi_1(f,u) = \sum_{a\in\mathbb{Z}_{p^2}\setminus p\mathbb{Z}_{p^2}}\sum_{x\in\Gamma_m} e^{2\pi i\frac{a(\mathrm{Tr}_m(f(x))+u)}{p^2}}.$$

Using Theorem 2.7, as $f(x)\in S(D)\setminus\{0\}$, we have

$$|\phi_1(f,u)| \le \frac{1}{p-1}p^{l_D+1}\left\lfloor\frac{p^{h_D}\frac{p^2-p}{2}(D-1)\left\lfloor 2p^{\frac{m}{2}-h_D}\right\rfloor}{p^{l_D+1}}\right\rfloor, \tag{3.22}$$

where l_D and h_D are as in Definition 2.9. By definition of $\theta_1(\boldsymbol{y})$ and $\phi_1(f,u)$, we also have

$$|\theta_1(\boldsymbol{y}) - \phi_1(f,u)| = \left|\sum_{l=0}^{p-1} e^{2\pi i\frac{(1+lp)(\mathrm{Tr}_m(f(0))+u)}{p^2}}\right| \le p. \tag{3.23}$$

Combining (3.22) and (3.23) we complete the proof. □

Remark 3.9. The maximal nontrivial correlation $\theta_{\max}(\mathcal{C}_D^1)$ of $\mathcal{C}_D^1$ is large. In fact even for any subset $S\subseteq\mathcal{C}_D^1$ with $\mathcal{F}_D^1\subsetneq S$, the maximal nontrivial correlation $\theta_{\max}(S)$ of S is at least $p(p^m-1)$. Indeed if $\mathcal{F}_D^1\subsetneq S\subseteq\mathcal{C}_D^1$, then there exist $(f(x),u_1)$, $(f(x),u_2)\in\widetilde{\mathcal{P}}_D^1$ with $u_2-u_1=jp$ and $1\le j\le p-1$ such that $\{s_1(i)\}_{i=0}^\infty = \{\Psi(\mathrm{Tr}_m(f(\beta^i))+u_1)\}_{i=0}^\infty$ and $\{s_2(i)\}_{i=0}^\infty = \{\Psi(\mathrm{Tr}_m(f(\beta^i))+u_2)\}_{i=0}^\infty$ are two cyclically distinct p-ary sequences in S. Then the modulus of the correlation between $\{s_1(i)\}_{i=0}^\infty$ and $\{s_2(i)\}_{i=0}^\infty$ at shift 0 is

$$\left|\sum_{l=0}^{p-1}\sum_{x\in\Gamma_m\setminus\{0\}} e^{2\pi i\frac{(1+lp)jp}{p^2}}\right| = (p^m-1)\left|e^{2\pi i\frac{j}{p}}\right|\cdot\left|\sum_{l=0}^{p-1}e^{2\pi i\frac{ljp}{p}}\right| = p(p^m-1).$$

Remark 3.10. For $p=2$, from $\mathcal{F}_D^1$ we retrieve the family of binary sequences $Q(D)$ of [H-K, section 8.8]. Let $\mathcal{F}_D^{1,0}$ be the subfamily of $\mathcal{F}_D^1$ defined as

$$\mathcal{F}_D^{1,0} = \left\{\{\Psi(\mathrm{Tr}_m(f(\beta^i)))\}_{i=0}^\infty : (f(x),0)\in\widehat{\mathcal{P}}_D^1\right\}.$$

Note that $\mathcal{F}_D^1$ is larger than $\mathcal{F}_D^{1,0}$ with the same upper bound on the maximal nontrivial correlation. For $p=2$, from $\mathcal{F}_D^{1,0}$ we obtain the family of binary sequences of [S-K-H].

Let $\mathcal{P}_D^2$ be the subset of $pS(D)_1\times(\mathbb{Z}_{p^2}\setminus p\mathbb{Z}_{p^2})$ defined as

$$\mathcal{P}_D^2 = \Big\{(pf(x),u)\in pS(D)_1\times(\mathbb{Z}_{p^2}\setminus p\mathbb{Z}_{p^2}) : \{\mathrm{Tr}_m(pf(\beta^i))\}_{i=0}^\infty \text{ has period } p^m-1\Big\}.$$

Using Proposition 3.6, we obtain that

$$|\mathcal{P}_D^2| = (p^2-p)\sum_{l|(p^m-1)}\mu(l)p^{m\left(\lfloor\frac{D}{l}\rfloor-\lfloor\frac{D}{pl}\rfloor\right)}, \tag{3.24}$$

where $\mu(\cdot)$ is the Möbius function.

For $(pf(x), u), (pg(x), v) \in \mathcal{P}_D^2$, we say $(pf(x), u)$ and $(pg(x), v)$ are *cyclically related* if there exist $0 \leq j \leq p-1$ and $0 \leq t \leq (p^m - 1) - 1$ such that $pg(x) = (1+p)^j(1-p)^t pf(\beta^t x)$ and $v = (1+p)^j(1-p)^t u$. Following the arguments similar to the ones for the case of $\mathcal{P}_D^1$, we observe that cyclically related elements of $\mathcal{P}_D^2$ form an equivalence relation and each equivalence class has $p(p^m-1)$ elements.

Let $\overline{\mathcal{P}}_D^2$ denote the set of equivalence classes in $\mathcal{P}_D^2$. We denote the equivalence class of $(pf(x), u) \in \mathcal{P}_D^2$ as $\overline{(pf(x), u)}$.

Let $(pf(x), u) \in \mathcal{P}_D^2$ be an element with $u = r_0(u) + pr_1(u), \quad r_0(u), r_1(u) \in \{0, 1, \ldots, p-1\}$. Let $1 \leq j \leq p-1$ be the integer with $jr_0(u) \equiv 1 \mod p$. Then $(1+p)^j pf(x) = (1+jp)pf(x) = pf(x)$ and $(1+p)^j u = (1+jp)(r_0(u)+pr_1(u)) = u+p$. Therefore we have

$$(pf(x), u) \in \mathcal{P}_D^2 \Rightarrow (pf(x), u+p) \in \overline{(pf(x), u)}. \tag{3.25}$$

From (3.9) and (3.25) we observe a different behavior of the equivalence classes in $\mathcal{P}_D^1$ and $\mathcal{P}_D^2$. Using (3.25) we choose a full set of representatives $\widetilde{\mathcal{P}}_D^2$ of the equivalence classes in $\overline{\mathcal{P}}_D^2$ such that

$$\widetilde{\mathcal{P}}_D^2 = \{(pf(x), u) \in \mathcal{P}_D^2 : \quad u \in \{1, \ldots, p-1\} \subseteq (\mathbb{Z}_{p^2} \setminus p\mathbb{Z}_{p^2})\}.$$

Let $\mathcal{F}_D^2$ be the family of p-ary sequences defined as

$$\mathcal{F}_D^2 = \{\{\Psi(\mathrm{Tr}_m(pf(\beta^i)) + u)\}_{i=0}^{\infty} : (pf(x), u) \in \widetilde{\mathcal{P}}_D^2\}.$$

THEOREM 3.11. *The family $\mathcal{F}_D^2$ has the following properties:*

(i) *The period of each sequence in $\mathcal{F}_D^2$ is $p(p^m-1)$.*

(ii) *The sequences in $\mathcal{F}_D^2$ are pairwise cyclically distinct.*

(iii) $|\mathcal{F}_D^2| = \frac{p-1}{p^m-1}\sum_{l|(p^m-1)} \mu(l) p^{m(\lfloor D/l \rfloor - \lfloor D/pl \rfloor)}$, *where $\mu(\cdot)$ is the Möbius function.*

(iv) *For the maximal nontrivial correlation $\theta_{\max}$ of $\mathcal{F}_D^2$, we have*

$$\theta_{\max} \leq \frac{1}{p-1} p^{l_D+1} \left\lfloor \frac{p^{h_D} \frac{p^2-p}{2}(D-1) \left\lfloor 2p^{\frac{m}{2}-h_D} \right\rfloor}{p^{l_D+1}} \right\rfloor + p,$$

where l_D and h_D are as in Definition 2.9.

Proof. Item (i) is clear. Next we prove items (ii) and (iii) together. Let $(pf(x), u) \in \widetilde{\mathcal{P}}_D^2$ with the corresponding p-ary sequence $\{s(i)\}_{i=0}^{\infty}$. We proceed as in the proof of Theorem 3.8. Assume that $0 \leq j \leq p-1$, $0 \leq t \leq (p^m-1)-1$ with the corresponding p-ary sequence $\{s_2(i)\}_{i=0}^{\infty}$ satisfying $s_2(i) = s(i + j(p^m-1) + t)$ for each $i \geq 0$. Let $(pg(x), u_2) \in \widetilde{\mathcal{P}}_D^2$ such that the p-ary sequence $\{\Psi(\mathrm{Tr}_m(pg(\beta^i)) + u_2)\}_{i=0}^{\infty}$ is $\{s_2(i)\}_{i=0}^{\infty}$. Note that $pf(x) = (1+p)^j(1-p)^t pf(x)$. Let $u_1 = (1+p)^j(1-p)^t u$ and let $\{s_1(i)\}_{i=0}^{\infty}$ be the corresponding p-ary sequence of $(pf(x), u_1) \in \mathcal{P}_D^2$. If either $j \neq 0$ or $t \neq 0$, then $(pf(x), u_1) \notin \widetilde{\mathcal{P}}_D^2$. As in the proof of Theorem 3.8 we have

$$\mathrm{Tr}_m(pf(\beta^i) + u_1) = \mathrm{Tr}_m(pg(\beta^i) + u_2) \tag{3.26}$$

for each $i \geq 0$. Then $\rho(u_1) = \rho(u_2)$ and hence $u_2 = u_1 + kp$, where $0 \leq k \leq p-1$. From (3.26) and Lemma 2.3 we obtain $pf(x) = pg(x) + kp$. By definition of $S(D)$ in (2.5), there is no monomial in $g(x)$ and $f(x)$ of degree zero with a nonzero coefficient. Therefore $k = 0$ and $(pf(x), u_1) = (pg(x), u_2)$, which completes the proof of item (ii). We complete the proof of item (iii) as in the proof of Theorem 3.8.

The proof of item (iv) is similar to the proof of Theorem 3.8 (iv). □

Now we give our main family of the p-ary sequences. Let $\mathcal{F}_D$ be the family of p-ary sequences defined as

$$\mathcal{F}_D = \mathcal{F}_D^1 \cup \mathcal{F}_D^2.$$

Theorem 3.12. *The family $\mathcal{F}_D$ has the following properties:*

(i) *The period of each sequence in $\mathcal{F}_D$ is $p(p^m-1)$.*

(ii) *The sequences in $\mathcal{F}_D$ are pairwise cyclically distinct.*

(iii)

$$|\mathcal{F}_D| = \frac{1}{p^m-1}\sum_{l|(p^m-1)} \mu(l) p^{m(\lfloor \frac{D}{l} \rfloor - \lfloor \frac{D}{p^2 l} \rfloor)} + \frac{p-2}{p^m-1}\sum_{l|(p^m-1)} \mu(l) p^{m(\lfloor \frac{D}{l} \rfloor - \lfloor \frac{D}{pl} \rfloor)},$$

where $\mu(\cdot)$ is the Möbius function.

(iv) *For the maximal nontrivial correlation $\theta_{\max}$ of $\mathcal{F}_D$, we have*

$$\theta_{\max} \leq \frac{1}{p-1} p^{l_D+1} \left\lfloor \frac{p^{h_D} \frac{p^2-p}{2}(D-1)\left\lfloor 2p^{\frac{m}{2}-h_D}\right\rfloor}{p^{l_D+1}} \right\rfloor + p,$$

where l_D and h_D are as in Definition 2.9.

Proof. Item (i) is clear. Note that any two distinct sequences from $\mathcal{F}_D^1$ (or from $\mathcal{F}_D^2$) are cyclically distinct. Moreover, if $(f(x),u) \in \widehat{\mathcal{P}}_D^1$, $(pg(x),v) \in \widetilde{\mathcal{P}}_D^2$ and $0 \leq j \leq p-1$, $0 \leq t \leq (p^m-1)-1$, then

$$\rho\left((1+p)^j(1-p)^t f(\beta^t x)\right) \neq 0 \text{ and } \rho\left((1+p)^j(1-p)^t pg(\beta^t x)\right) = 0.$$

We complete the proof of item (ii) using Lemmas 2.3 and 2.5 as in the proof of Theorem 3.8 (ii).

Now we prove item (iii). Let $\{s_1(i)\}_{i=0}^{\infty}$ and $\{s_2(i)\}_{i=0}^{\infty}$ be the p-ary sequences of $\mathcal{F}_D^1$ and $\mathcal{F}_D^2$ obtained from $(f(x),u) \in \widehat{\mathcal{P}}_D^1$ and $(pg(x),v) \in \widetilde{\mathcal{P}}_D^2$, respectively. If $s_1(i) = s_2(i)$ for each $i \geq 0$, then

$$r_0(\mathrm{Tr}_m(f(\beta^i)) + u) = r_0(p\mathrm{Tr}_m(g(\beta^i)) + v). \tag{3.27}$$

Using (3.27) and Lemmas 2.3 and 2.5 we obtain that $f(x) = 0$, which is a contradiction.

We prove item (iv) using the methods of the proof of Theorem 3.8 (iv). □

Note that $\mathcal{F}_D$ is larger than $\mathcal{F}_D^1$ while the sequences in them have the same period and the same upper bound for their maximal nontrivial correlation in Theorems 3.8 and 3.12.

Example 3.13. In this example we assume that $p = 2$. We recall that the subfamily $\mathcal{F}_D^{1,0}$ of $\mathcal{F}_D^1$ given in Remark 3.10 corresponds to the family of sequences in [S-K-H] and $\mathcal{F}_D^1$ corresponds to the family of sequences in [H-K, section 8.8]. For $\mathcal{F}_D^{1,0}$, $\mathcal{F}_D^1$, and $\mathcal{F}_D$, we have the same period length and the same upper bounds for their maximal nontrivial correlation. The size of the family $\mathcal{F}_D^1$ is twice the size of the family $\mathcal{F}_D^{1,0}$. In [S-K-H, Table 2], for small values of D, the upper bounds for the maximal nontrivial correlation of $\mathcal{F}_D^{1,0}$ are given. In Table 1, we compare the family sizes of $\mathcal{F}_D^1$ and $\mathcal{F}_D$ for small values of D and m.

Table 1
Comparison of family sizes of $\mathcal{F}_D^1$ and $\mathcal{F}_D$ for $p = 2$.

m	D	$\lvert\mathcal{F}_D^1\rvert$	$\lvert\mathcal{F}_D\rvert$
3	2	8	9
5	2	32	33
7	2	128	129
5	3	1024	1057
7	3	16384	16513
5	5	32768	33825
7	5	2097152	2113665

Table 2
Comparison of a family for $p = 3$ with some families for $p = 2$ from Theorem 3.12.

p	D	m	L =period	$\log S/\log L$	$\theta^*_{\max}/\sqrt{L}$
3	8	9	59049	6.30003...	12.06820...
2	8	21	4194302	4.77272...	9.84472...
2	9	17	262142	5.66667...	11.25394...
2	10	15	65534	6.56252...	12.63300...
2	11	13	16382	7.42867...	13.76646...

Note that the family sizes of the binary sequences in [U-S] are about half of the family sizes of the binary sequences obtained from Theorem 3.12 for $D = 2$. For $D = 3$, the periods of the binary sequence families match only for one of the families in [B], which is the family noted as twice shortened Delsarte–Goethals codes in the table of [B]. For that family, the sizes are similar while the maximal nontrivial correlation upper bound is better for the sequences from Theorem 3.12. For $D \geq 3$ and $D \geq 5$, the binary sequence families from Theorem 3.12 are much larger than the ones from [U-S] and [B], respectively.

Example 3.14. In this example we compare a sequence family for $p = 3$ with the relevant sequence families for $p = 2$, where the sequences are obtained using Theorem 3.12. For $p = 3$, $D = 8$, and $m = 9$, we get a sequence family of size S and of period $L = 59049$ such that

$$\frac{\log S}{\log L} = 6.30003\ldots, \quad \frac{\theta_{\max}}{\sqrt{L}} \leq 12.06820\ldots.$$

In Table 2, S denotes the family size and $\theta^*_{\max}$ denotes the upper bound on the maximal nontrivial correlation $\theta_{\max}$ of the corresponding sequence families from Theorem 3.12. For $p = 2$ and D in Table 2, m is chosen to be the smallest positive odd integer such that the corresponding family size is at least the size of the sequence family for $p = 3$. We observe that the parameters of the sequence family for $p = 3$ are comparable to the parameters of the other sequence families in Table 2. In particular there is no sequence family for $p = 2$ in Table 2 such that $\log S/\log L$ is larger than that of $p = 3$ and $\theta^*_{\max}/\sqrt{L}$ is smaller than that of $p = 3$ simultaneously.

Remark 3.15. Let n be a positive divisor of $p^m - 1$ and let $\zeta = \beta^{\frac{p^m-1}{n}}$ be a primitive nth root of unity. Changing β with ζ and putting a suitable condition on D, we obtain p-ary codes of length pn and p-ary sequence families of period pn in an analogous way. Moreover, using the methods of this paper, we can also estimate the minimum distance of such codes and the maximum nontrivial correlation of such sequence families.

4. Conclusion. In this paper, a family of codes over $\mathbb{F}_p$ and several families of pairwise cyclically distinct p-ary sequences of period $p(p^m-1)$ of low correlation have been constructed as the Gray image and generalized Nechaev–Gray images, respectively, of some trace codes over $\mathbb{Z}_{p^2}$. The $\mathbb{F}_p$-codes, mostly nonlinear, are of length p^{m+1} and size $p^2 \cdot p^{m\left(D-\lfloor D/p^2\rfloor\right)}$, where $1 \leq D \leq p^{m/2}$. A lower bound for their minimum distance is obtained through the bound of [L-O]. The sequences compare favorably with certain known p-ary sequences of period p^m-1 (cf. [H-K, Table 4]). In fact, even in the case $p=2$, one of these families is slightly larger than the family $Q(D)$ of [H-K, section 8.8], while they share the same period and the same bound for the maximum nontrivial correlation.

Acknowledgments. The authors are thankful to the anonymous referees and the editor for their helpful comments.

REFERENCES

[B] A. Barg, *On Small Familes of Sequences with Low Periodic Correlation*, Lecture Notes in Comput. Sci. 781, Springer-Verlag, Berlin, 1994, pp. 154–158.

[C] C. Carlet, *$\mathbb{Z}_{2^k}$-linear codes*, IEEE Trans. Inform. Theory, 44 (1998), pp. 1543–1547.

[C-H] I. Constantinescu and W. Heise, *A metric for codes over residue class rings of integers*, Probl. Inf. Transm., 33 (1997), pp. 208–213.

[G-S] M. Greferath and S. E. Schmidt, *Gray isometries for finite chain rings and a nonlinear ternary $(36, 3^{12}, 15)$ code*, IEEE Trans. Inform. Theory, 45 (1999), pp. 2522–2524.

[H-K] T. Helleseth and P. V. Kumar, *Sequences with low correlation*, in Handbook of Coding Theory, Vol. I, II, V. S. Pless and W. C. Huffman, eds., North-Holland, Amsterdam, 1998, pp. 1765–1853.

[K-H-C] P. V. Kumar, T. Helleseth, and A. R. Calderbank, *An upper bound for Weil exponential sums over Galois rings with applications*, IEEE Trans. Inform. Theory, 41 (1995), pp. 456–468.

[H-K-M-S] T. Helleseth, P. V. Kumar, O. Moreno, and A. G. Shanbhag, *Improved estimates via exponential sums for the minimum distance of $\mathbb{Z}_4$-linear trace codes*, IEEE Trans. Inform. Theory, 42 (1996), pp. 1212–1216.

[N] A. A. Nechaev, *The Kerdock code in a cyclic form*, Discrete Math. Appl., 1 (1991), pp. 365–384.

[L-N] R. Lidl and H. Niederreiter, *Finite Fields*, Cambridge University Press, Cambridge, UK, 1997.

[L-B] S. Ling and J. T. Blackford, *$\mathbb{Z}_{p^{k+1}}$-linear codes*, IEEE Trans. Inform. Theory, 48 (2002), pp. 2592–2605.

[L-O] S. Ling and F. Özbudak, *An improvement on the bounds of Weil exponential sums over Galois rings with some application*, IEEE Trans. Inform. Theory, 50 (2004), pp. 2529–2539.

[L-S] S. Ling and P. Solé, *Nonlinear p-ary sequences*, Appl. Algebra Engrg. Comm. Comput., 14 (2003), pp. 117–125.

[U-S] P. Udaya and M. U. Siddiqi, *Optimal biphase sequences with large linear complexity derived from sequences over $\mathbb{Z}_4$*, IEEE Trans. Inform. Theory, 42 (1996), pp. 206–217.

[S-K-H] A. G. Shanbhag, P. V. Kumar, and T. Helleseth, *Improved binary codes and sequence families from $\mathbb{Z}_4$-linear codes*, IEEE Trans. Inform. Theory, 42 (1996), pp. 1582–1587.

SIAM J. DISCRETE MATH.
Vol. 19, No. 4, pp. 1029–1055

SPARSE DISTANCE PRESERVERS AND ADDITIVE SPANNERS*

BÉLA BOLLOBÁS[†], DON COPPERSMITH[‡], AND MICHAEL ELKIN[§]

Abstract. For an unweighted graph $G = (V, E)$, $G' = (V, E')$ is a subgraph if $E' \subseteq E$, and $G'' = (V'', E'', \omega)$ is a *Steiner graph* if $V \subseteq V''$, and for any pair of vertices $u, w \in V$, the distance between them in G'' (denoted $d_{G''}(u, w)$) is at least the distance between them in G (denoted $d_G(u, w)$).

In this paper we introduce the notion of *distance preserver*. A subgraph (resp., Steiner graph) G' of a graph G is a subgraph (resp., Steiner) *D-preserver* of G if for every pair of vertices $u, w \in V$ with $d_G(u, w) \geq D$, $d_{G'}(u, w) = d_G(u, w)$. We show that any graph (resp., digraph) has a *subgraph* D-preserver with at most $O(n^2/D)$ edges (resp., arcs), and there are graphs and digraphs for which any *undirected Steiner* D-preserver contains $\Omega(n^2/D)$ edges. However, we show that if one allows a *directed Steiner* (*diSteiner*) D-preserver, then these bounds can be improved. Specifically, we show that for any graph or digraph there exists a diSteiner D-preserver with $O(\frac{n^2 \cdot \log D}{D \cdot \log n})$ arcs, and that this result is tight up to a constant factor.

We also study *D-preserving* distance labeling schemes, that are labeling schemes that guarantee precise calculation of distances between pairs of vertices that are at a distance of at least D one from another. We show that there exists a D-preserving labeling scheme with labels of size $O(\frac{n}{D} \log^2 n)$, and that labels of size $\Omega(\frac{n}{D} \log D)$ are required for any D-preserving labeling scheme.

Key words. graph theory, spanners, distance preservation

AMS subject classifications. 05C12, 05C85, 68R05

DOI. 10.1137/S0895480103431046

1. Introduction.

A graph $G' = (V, E')$ is a *subgraph* of an unweighted graph $G = (V, E)$ if $E' \subseteq E$. The *distance* from a vertex u to a vertex w in G, denoted $d_G(u, w)$, is the number of edges in the shortest (in terms of the number of edges) path from u to w in G. Note that the distances in a subgraph G' are no smaller than the corresponding distances in G. A (possibly weighted) graph $G' = (V', E', \omega)$ is a *Steiner graph* of G if $V \subseteq V'$, and for any pair of vertices $u, w \in V$, $d_{G'}(u, w) \geq d_G(u, w)$, and for any edge $e' \in E'$, $\omega(e') \geq 0$. Observe that any subgraph G' of G is, in particular, a Steiner graph of G. A subgraph or a Steiner graph G' of G that *approximates* (in some sense) all the distances in G is called a *spanner*. In particular, for a positive integer parameter κ, G' is a *κ-spanner* of G, if for any pair of vertices u, w in G, $d_{G'}(u, w) \leq \kappa \cdot d_G(u, w)$. The number κ is called the *stretch* or *distortion* factor of the spanner G'.

Spanners were intensively studied during the last fifteen years. They have multiple applications in distributed computing [2, 3, 21, 14, 4] and computational geometry

*Received by the editors July 8, 2003; accepted for publication (in revised form) August 1, 2005; published electronically February 3, 2006. A preliminary version of this paper appeared in SODA 2003 [6].

http://www.siam.org/journals/sidma/19-4/43104.html

[†]Microsoft Research, One Microsoft Way, Redmond, WA 98122; Department of Mathematical Sciences, University of Memphis, Memphis, TN 38152 (bollobas@msci.memphis.edu); Trinity College, Cambridge CB2 1TQ, UK. This author's research was supported by NSF grant DSM 9971788 and DARPA grant F33615-01-C-190.

[‡]IBM Research, Yorktown Heights, NY 10598 (dcopper@us.ibm.com).

[§]Department of Computer Science, Ben-Gurion University of the Negev, Israel (elkinm@cs.bgu.ac.il). Part of this work was done at the School of Mathematics, Institute for Advanced Study, Princeton, NJ 08540.

[10, 13]. Furthermore, constructing a spanner and applying existing algorithms on it was used as an algorithmic technique in [4, 11, 12, 14].

Peleg and Schäffer [20] have shown that for any positive integer κ and any n-vertex graph G there exists a subgraph $O(\kappa)$-spanner G' with $O(n^{1+1/\kappa})$ edges. Note that this result indicates a *tradeoff* between the stretch of the spanner and the number of edges it uses. This tradeoff was shown to be essentially the best possible in [20], but some constant factors were improved later on in [1, 9]. These papers also generalized the result to *weighted* graphs. Recently, Elkin and Peleg [15, 14] have shown that the aforementioned tradeoff is tight only as far as the distortion of *small distances* is considered, and can be almost eliminated whenever one is interested in approximating the distances that are greater than a certain constant. Specifically, it is shown there that for any pair of parameters $\epsilon > 0$, $\kappa = 1, 2, \ldots$ there exists a threshold $\beta = \beta(\epsilon, \kappa)$ such that for any n-vertex graph G there exists a subgraph spanner G' with $O(n^{1+1/\kappa})$ edges such that for any pair of vertices u, w that are at a distance of at least β one from another in G, the distance in G' is at most by a factor $1 + \epsilon$ greater than the one in G (i.e., $d_{G'}(u, w) \le (1 + \epsilon) \cdot d_G(u, w)$). In other words, *large distances* can be approximated arbitrarily well by arbitrarily sparse spanners. In view of this result due to [15], it is natural to ask whether approximation is at all necessary whenever large distances are under consideration, or, maybe large distances can be *preserved* using a sparse spanner.

To address this question, we introduce a notion of a *distance preserving subgraph*, briefly, a *preserver.* A subgraph G' of a graph G is a D*-preserver* of G if for every pair of vertices $u, w \in V$ with $d_G(u, w) \ge D$, $d_{G'}(u, w) = d_G(u, w)$. (The same definition applies to Steiner graphs as well.) We show that any graph (resp., digraph) has a *subgraph* D-preserver with at most $O(n^2/D)$ edges (resp., arcs), and there are graphs and digraphs for which any *undirected Steiner* D-preserver contains $\Omega(n^2/D)$ edges (resp., arcs). However, we show that if one allows a *directed Steiner* (*diSteiner*) D-preserver, then these bounds can be improved. Specifically, we show that for any graph or digraph there exists a diSteiner D-preserver with $O(\frac{n^2 \cdot \log D}{D \cdot \log n})$ arcs, and that this result is tight up to a constant factor. In particular, it follows that for any graph or digraph there is a diSteiner 1-preserver with $O(n^2/\log n)$ arcs. Generalizing this result, we show that for any graph (resp., digraph) with $m \ge c' \cdot n^{3/2}$ edges (resp., arcs), for some small constant $c' > 1$, there is a diSteiner 1-preserver with fewer than m arcs, and that a factor of $\frac{\log n}{c \log \log n}$ (resp., $\log^{1-\gamma} n$) can be "saved" for $m = n^2/\log^c n$ (resp., $m = n^2/2^{\log^\gamma n}$) for any $c > 0$ (resp., $0 < \gamma < 1$). We also show that for any bipartite graph with m edges and girth greater than 4, any diSteiner 1-preserver contains at least m arcs, and as there are such graphs with $m = (1/2 + o(1))n^{3/2}$ edges, it follows that this upper bound cannot be generalized to graphs with $m \le (1/2)n^{3/2}$ edges.

Our proof of the existence of sparse diSteiner preservers uses the following theorem.

THEOREM 1.1 (cf. [5]). *Let G be an n-vertex graph with average degree d, and s and t be positive integers such that $s \le t$ and $n\binom{d}{t} > (s-1)\binom{n}{t}$. Then G contains a $K_{s,t}$ (complete bipartite subgraph with one bipartition of size s and another of size t).*

In order to convert our proof of *existence* of diSteiner D-preservers into a polynomial time algorithm for computing them, we devised a constructive proof of Theorem 1.1. This proof might be of independent interest in the context of Ramsey theory. From an algorithmic perspective, this proof may serve as an algorithm for computing a subgraph isomorphic to $K_{s,t}$ in a graph that satisfies the assumptions of Theorem 1.1. The complexity of this algorithm is $O(n^2 \cdot t)$. We use this result for devising an algorithm with a running time of $O(n^4 \frac{(\log\log n)^2}{\log n})$ (resp., $O(m^2 \cdot n)$) for computing a diSteiner 1-preserver (resp., D-preserver) with $O(n^2/\log n)$ (resp., $O(\frac{n^2 \log D}{D \cdot \log n})$) arcs for an arbitrary n-vertex graph with m edges. We remark that any improvement of a factor of $\Omega(n)$ in the running time of an algorithm for constructing a diSteiner 1-preserver would have some interesting applications to efficient computation of distances in dense graphs (by computing their diSteiner 1-preserver, and performing distance computations on the 1-preserver, assuming that the latter is sparser than the original graph).

In particular, our results address the aforementioned question and show that *approximation* of large distances is indeed necessary as far as arbitrarily sparse spanners are considered, as there exist infinite families of graphs in which large distances cannot be preserved by a *sparse* spanner.

We also generalize the definition of D-preserver, and say that G' is a (D, g)-*preserver* of G if for any pair of vertices $u, w \in V$ such that $d_G(u, w) \geq D$, we have $d_{G'}(u, w) \leq d_G(u, w) + g$. In this context, we show upper and lower bounds on the maximal number m_1 of edges in a graph for which any subgraph (D, g)-preserver contains at least m_1 edges. We show that $\Omega(\frac{n^{1+c_0/(g+2)}}{g \cdot D^{c_0/(g+2)}}) = m_1 = O(\frac{n^{1+1/\lfloor g/4 \rfloor}}{D^{1/\lfloor g/4 \rfloor}})$, where $4/3 \leq c_0 \leq 2$, and under the Erdős girth conjecture, $c_0 = 2$. The lower bound serves also as a lower bound on the minimal number m_2 such that any graph has a subgraph (D, g)-preserver with m_2 edges. However, so far we are not able to prove a non-trivial upper bound on the size of (D, g)-preservers, and, in particular, it is not clear to us whether these two dual notions m_1 and m_2 are equal.

We also study the problem of preserving long distances in the context of *distance labeling schemes*. Distance labeling scheme is a pair of functions $(\mathcal{M}, \mathcal{D})$. The *labeling* function $\mathcal{M}$, given a graph G and a vertex v, returns a bit string, often called the *label of* v. The *query-answering* function $\mathcal{D}$, given a pair of labels, returns an estimate of the distance between the corresponding pair of vertices.

The problem of devising distance labeling schemes with *short* labels was introduced in [19], and has been intensively studied since then [17, 24, 23]. We consider D-*preserving* labeling schemes, that are schemes that satisfy $\mathcal{D}(\mathcal{M}(G, u), \mathcal{M}(G, w)) = d_G(u, w)$ for any graph $G = (V, E)$ and pair of vertices $u, w \in V$ such that $d_G(u, w) \geq D$. We show that there exists a D-preserving labeling scheme with labels of size $O(\frac{n}{D} \log^2 n)$, and that labels of size $\Omega(\frac{n}{D} \log D)$ are required for any D-preserving labeling scheme.

Related work. We remark that our results on distance preservers that are presented in this paper were used by us in [6] to derive the first nontrivial bounds for arbitrarily sparse *additive spanners*. Specifically, in [6] we devise a construction of additive $O(2^{1/\delta} n^{(1-\delta)\frac{\lceil 1/\delta \rceil - 2}{\lceil 1/\delta \rceil - 1}})$-spanners with $O(n^{1+\delta})$ edges for any graph and any $\delta > 0$. In particular, this implies a construction of additive $O(n^{4/9+(2/3)\epsilon})$-spanners with $O(n^{4/3-\epsilon})$ edges. In the consequent to this paper, these bounds were improved by Baswana et al. [7].

After our basic results (the existence of subgraph D-preserver with $O(n^2/D)$ edges and the lower bound of $\Omega(n^2/D)$ on the number of edges in subgraph D-preservers) were communicated to Mikkel Thorup, he devised [22] a more efficient randomized procedure for computing a subgraph D-preserver of size $O(n^2 \log n/D)$ (greater than optimal by a logarithmic factor). This more efficient procedure uses some techniques of [25] from the area of dynamic algorithms. The efficiency of the procedure of [22] makes it more suitable for algorithmic applications such as (and this is, indeed, the motivation of [22]) computing shortest paths between pairs of vertices that are at distance at least D one from another. We use a similar idea to devise D-preserving labeling schemes.

Our algorithm for constructing sparse diSteiner 1-preservers for general graphs successively extracts large bipartite cliques and replaces them by directed stars. A similar idea of extracting large bipartite cliques was used by Feder and Motwani in [16] for constructing *compressions* of graphs. The notion of compression graph is somewhat similar to the notion of Steiner graph, but the distances in a compression graph may be *shorter* than the distances in the original graph.

Structure of the paper. In section 2 we discuss the issue of distance preservation, which is the main topic of this paper. This section is divided into subsection 2.2, which is devoted to the lower bounds, and subsection 2.3, which is devoted to the upper bounds. In section 2.3.2 we address the algorithmic aspects of our paper. In particular, this section contains our constructive proof of Theorem 1.1 and a description of a D-preserving labeling scheme.

2. Distance preservation.

2.1. Discussion.

A subgraph G' of a graph $G = (V, E)$ is its (α, β)-*spanner* if for any pair of vertices $u, w \in V$, $d_{G'}(u, w) \leq \alpha \cdot d_G(u, w) + \beta$. Our starting point is the following result from [15].

THEOREM 2.1 (see [15]). *Given constants $0 < \epsilon, \delta < 1$, there is a constant $\beta = \beta(\delta, \epsilon) = (1/\delta)^{\max\{\log\log 1/\delta - \log \epsilon)(1 - 1/\log 1/\delta), 3\}}$ such that for any graph G, there exists a constructible in polynomial time $(1+\epsilon, \beta)$-spanner $G' = (V, E')$ and Steiner $(1+\epsilon, \beta)$-spanner $G'' = (V'', E'', \omega)$ with $|E'| = O(\beta n^{1+\delta})$ and $|E''| = O(n^{1+\delta})$.*

(The result about Steiner spanners is implicit in [15].)

Note that Theorem 2.1 implies that for any fixed $\epsilon, \delta > 0$ there exists fixed $\beta' = \beta'(\delta, \epsilon)$ such that for any undirected graph $G = (V, E)$ there exists a subgraph $G' = (V, E')$, $E' \subseteq E$ with $|E'| = O(n^{1+\delta})$ edges that *approximates within a multiplicative factor of* $1+\epsilon$ all the distances that are already greater than β'. We start with showing that this result is optimal in the sense that $(1 + \epsilon)$-approximation is necessary, and, furthermore, for any fixed $\delta > 0$ there is no fixed $\beta' = \beta'(\delta)$ such that for any undirected graph $G = (V, E)$ there exists a subgraph $G' = (V, E')$, $E' \subseteq E$ with $|E'| = O(n^{1+\delta})$ edges that *preserves* all the distances already greater than β'.

To facilitate the discussion, let us introduce some definitions.

DEFINITION 2.2. *For an integer $D \geq 1$, a subgraph $G' = (V, E')$ of a graph $G = (V, E)$ is said to be a* (subgraph) D-preserver *of G, if for any pair of vertices $u, w \in V$ with $d_G(u, w) \geq D$, $d_{G'}(u, w) = d_G(u, w)$.*

The definition extends in a natural way to *Steiner D-preservers*.

DEFINITION 2.3. *For integer numbers $n \geq 2$ and $1 \leq D \leq n - 1$, let $f(D, n)$ (resp., $f_S(D, n)$) be the minimal number such that for any n-vertex graph there exists a subgraph (resp., Steiner) D-preserver with at most $f(D, n)$ (resp., $f_S(D, n)$) edges. Also, let $\bar{f}(D, n)$ (resp., $\bar{f}_S(D, n)$) be the maximal number m of edges in an n-vertex graph whose any subgraph (resp., Steiner) D-preserver contains at least m edges.*

On *directed* graphs, let $f^{dir}(D, n)$, $\bar{f}^{dir}(D, n)$, $f_S^{dir}(D, n)$, and $\bar{f}_S^{dir}(D, n)$ denote the corresponding quantities.

The equality between these dual notions follows from their definitions.

LEMMA 2.4. *For integer numbers $n \geq 2$ and $1 \leq D \leq n-1$, $f(D, n) = \bar{f}(D, n)$.*

Proof. By definition of $\bar{f}(D,n)$, there exists an n-vertex graph G_0 with $\bar{f}(D,n)$ edges whose any D-preserver contains at least $\bar{f}(D,n)$ edges. By definition of $f(D,n)$, for any n-vertex graph G, there exists a D-preserver with at most $f(D,n)$ edges. In particular, there is a D-preserver of G_0 with $m' \le f(D,n)$ edges. As $m' \ge \bar{f}(D,n)$, it follows that $\bar{f}(D,n) \le f(D,n)$.

For the opposite direction, note that by the definition of $f(D,n)$, there exists an n-vertex graph $G_1 = (V_1, E_1)$ such that any D-preserver of G_1 contains at least $f(D,n)$ edges, and at least one of them contains precisely $f(D,n)$ edges. Consider the D-preserver G_1' of G_1 that contains precisely $f(D,n)$ edges. For any pair of vertices $u, w \in V_1$ such that $d_{G_1}(u,w) \ge D$, $d_{G_1'}(u,w) = d_{G_1}(u,w)$. Consider some subgraph $G_1'' = (V_1, E_1'')$ of G_1' such that E_1'' is a strict subset of E_1' (i.e., $E_1'' \subset E_1'$). As $|E_1''| < |E_1'| = f(D,n)$, it follows that G_1'' is not a D-preserver of G_1. That is, there is a pair of vertices $u, w \in V_1$ such that $d_{G_1}(u,w) \le D$, but $d_{G_1''}(u,w) > d_{G_1}(u,w) = d_{G_1'}(u,w)$. Hence, G_1'' is not a D-preserver of G_1' as well. Hence any D-preserver of G_1' contains at least $f(D,n)$ edges. As $\bar{f}(D,n)$ is the maximal number of edges in a graph whose any D-preserver contains at least the same number of edges as the graph itself, it follows that $f(D,n) \le \bar{f}(D,n)$. This concludes the proof. □

Analogously, $f_S(D,n) = \bar{f}_S(D,n)$, $f^{dir}(D,n) = \bar{f}^{dir}(D,n)$ and $f_S^{dir}(D,n) = \bar{f}_S^{dir}(D,n)$. Also, as any subgraph D-preserver is, in particular, a Steiner D-preserver, it follows that $f_S(D,n) = \bar{f}_S(D,n) \le f(D,n) = \bar{f}(D,n)$, and $f_S^{dir}(D,n) = f_S^{dir}(D,n) \le f^{dir}(D,n) = \bar{f}^{dir}(D,n)$.

2.2. Lower bounds.

2.2.1. Undirected graphs. The following example shows that for $0 < \delta < 1$ there is no fixed $D = D(\delta)$ such that for any undirected n-vertex graph G there exists a D-preserver G' with $O(n^{1+\delta})$ edges. Consider a clique of $n^{1/2+\delta/2}$ vertices. (In most of the cases we ignore the issue of a possible nonintegrality of different quantities; anyway this affects only the lower order terms), with a path of length $D = n^{1/2-\delta/2}$ attached to every vertex. Denote this graph by $G_0 = (V_0, E_0)$.

DEFINITION 2.5. *Given a digraph (resp., undirected graph) $G = (V,E)$, a sequence of vertices $P = (v_0, v_1, \ldots, v_s)$, $s \ge 0$, is called a* walk *if $\langle v_i, v_{i+1}\rangle$ (resp., (v_i, v_{i+1})) belongs to E, for every integer i, $0 \le i \le s-1$. A walk $P = (v_0, v_1, \ldots, v_s)$ is a* path, *if $v_i \ne v_j$ for all integer i, j, $0 \le i, j \le s$, $i \ne j$.*

LEMMA 2.6. *$f(D,n) = \bar{f}(D,n) = \Omega(n^2/D^2)$.*

Proof. Let $W = \{w_1, w_2, \ldots, w_{n/D}\}$ be the set of the vertices of the clique, and $U = \{u_1, u_2, \ldots, u_{n/D}\}$ be the set of the endpoints of the paths that do not belong to the clique. (Throughout this section it is assumed that D divides n.) Assume also that w_i's and u_i's are ordered in such a way that for any $i = 1, 2, \ldots, n/D$, w_i and u_i are two endpoints of the same path of length D.

Note that $|E_0| = \Theta(n^{1+\delta}) = \Theta(n^2/D^2)$; see Figure 1. Also, observe that no strict subgraph of G_0 may serve as a D-preserver for G_0. This is because removing an edge from one of the paths makes the graph disconnected. In particular, in this case the distance between the nonclique endpoint of the path from which the edge was removed, and an endpoint of some other path, becomes infinity, and it is $2D-1 \ge D$ in G_0. Also, removal of some clique edge (w_i, w_j), $i \ne j$, $i, j = 1, 2, \ldots, n/D$ results in increasing the distance between u_i and u_j. Note that $d_{G_0}(u_i, u_j) \ge 2D$. Hence, $\bar{f}(D,n) = \Omega(n^2/D^2)$. Therefore, by Lemma 2.4, $f(D,n) = \Omega(n^2/D^2)$. □

Note that $f(D,n) = \Omega(n^2/D^2)$ and $f(D,n) = O(n^{1+\delta})$ implies $D = \Omega(n^{1/2-\delta/2})$. In other words, for any $0 < \delta < 1$, there are n-vertex graphs for which any subgraph with $O(n^{1+\delta})$ edges is not a D-preserver for any $D = o(n^{1/2-\delta/2})$.

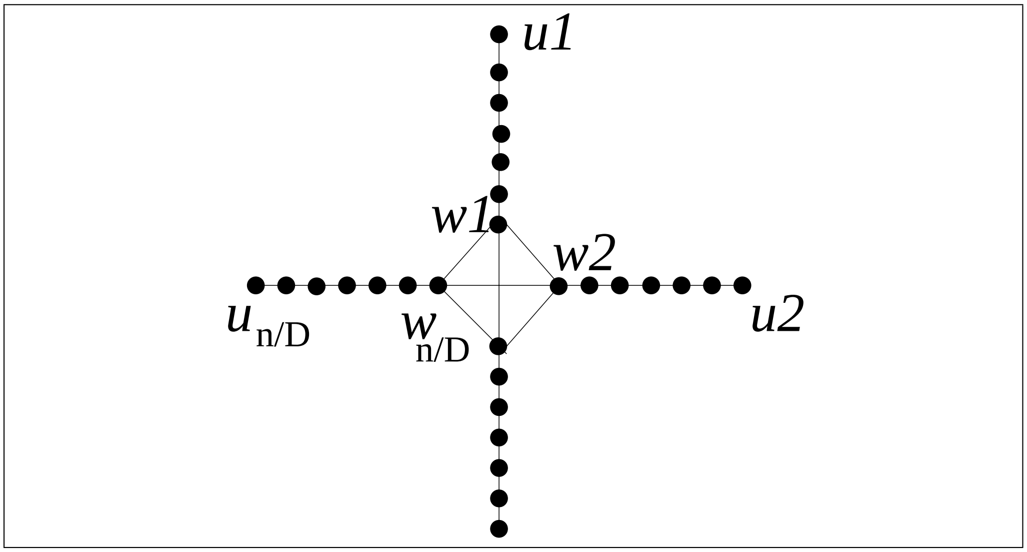


FIG. 1. *A clique of size n/D between vertices $w_1, w_2, \ldots, w_{n/D}$, with a path of length D attached to each w_i, that connects it to u_i. This example implies that $f(D, n) = \Omega(n^2/D^2)$.*

Note, however, that the graph G_0 does admit a *Steiner* 1-preserver of linear size. In this Steiner graph $V_0' = V_0 \cup \{s\}$, and the clique of size n/D in G_0 is replaced in G_0' by a star rooted in the new vertex s. All the edges of this star are of weight 1/2. The paths remain unchanged.

Next, we show that

$$\bar{f}_S(D, n) \geq n^2/4D. \tag{1}$$

This improves the lower bound of Lemma 2.6 in two respects. First, this lower bound applies to *Steiner* D-preservers, while the lower bound of Lemma 2.6 applies only to subgraph D-preservers. Second, this lower bound is stronger by a factor of $\Theta(D)$ than that of Lemma 2.6.

Consider the following example. Let $G_1 = (V_1, E_1)$ be an $n/2 \times n/2D$ complete bipartite graph between the vertex sets $X = \{x_1, x_2, \ldots, x_{n/2}\}$ and $Y = \{y_1, y_2, \ldots, y_{n/2D}\}$ with paths of length $(D - 1)$ attached to each y_i, that connect y_i with z_i for $i = 1, 2, \ldots, n/2D$. It is easy to see that the only subgraph D-preserver of G_1 is G_1 itself. As the graph contains $|E| \geq n^2/4D$ edges, a lower bound of $f(D, n) = \bar{f}(D, n) \geq n^2/4D$ follows. Let $\vec{G}_1$ be the digraph obtained by replacing every edge of G_1 by two arcs, one in each direction. As the only subgraph D-preserver of $\vec{G}_1$ is $\vec{G}_1$, a lower bound on $f^{dir}(D, n)$ follows:

$$f^{dir}(D, n) = \bar{f}^{dir}(D, n) \geq n^2/2D. \tag{2}$$

However, the analogous lower bound for Steiner D-preservers applies only to the undirected case and requires a more delicate treatment (it is easy to see that $\vec{G}_1$ admits a *directed* Steiner 1-preserver with a linear number of edges).

Consider an (undirected) Steiner D-preserver $G_1' = (V_1', E_1', \omega)$ of G_1. Assume, without loss of generality, that $\omega(e) > 0$ for an edge $e \in E'$. (Recall that by the definition of a Steiner graph, $\omega(e) \geq 0$.) Indeed, consider an edge $e = (u, w)$ such that

$\omega(e) = 0$. First, note that either $u \in V_1' \setminus V_1$ or $w \in V_1' \setminus V_1$ (or both of them). This is because if $u, w \in V_1$, then $d_{G_1'}(u, w) \geq d_{G_1}(u, w)$, by the definition of a Steiner graph. Therefore, the edge (u, w) can be contracted (and if one of the vertices belongs to V_1, then the other one is eliminated) without changing the distances between the pairs of vertices $s, t \in V_1$.

In addition, for every pair $(i, j) \in \{1, 2, \ldots, n/2\} \times \{1, 2, \ldots, n/2D\}$, let us associate a shortest path $P_{i,j}$ between x_i and z_j in G_1'.

For some fixed index $i \in \{1, 2, \ldots, n/2\}$, consider the set of paths $\{P_{ij} \mid j \in \{1, 2, \ldots, n/(2D)\}$. Let e_j denote the edge of P_{ij} that is adjacent to x_i.

LEMMA 2.7. *For each pair of distinct indices j, ℓ, $1 \leq j, \ell \leq n/(2D)$, $e_j \neq e_\ell$.*

Proof. Suppose for contradiction that for some pair of distinct indices $j, \ell \in \{1, 2, \ldots, n/(2D)\}$, $e_j = (x_i, w_j) = e_\ell$. Observe that by triangle inequality, $d_{G_1'}(z_j, z_\ell) \leq d_{G_1'}(w_j, z_j) + d_{G_1'}(w_j, z_\ell)$. Also, as w_j lies on the shortest path P_{ij} between x_i and Z_j, it follows that $d_{G_1'}(w_j, z_j) = d_{G_1'}(x_i, z_j) - \omega((x_i, w_j)) < D$.

Analogously, as w_j lies on the shortest path $P_{i\ell}$ between X_i and z_j, it follows that $d_{G_1'}(w_j, z_\ell) < D$. Consequently, $D_{G_1'}(z_j, z_\ell) < 2D = d_{G_1}(z_j, z_\ell)$—a contradiction.

Consider the edge set $H_i = \{e_j \mid j \in \{1, 2, \ldots, n/(2D)\}$. It follows that $|H_i| = n/(2D)$. Observe also that for two distinct indices $i, k \in \{1, 2, \ldots, n/2\}$, the edge sets H_i and H_k are disjoint. (As each edge of H_i is adjacent to the vertex x_i, and each vertex of H_k is adjacent to the vertex x_k, and $i \neq k$. It is also easy to see that the edge (x_i, x_k) cannot belong to P_{ij} or P_{kj}.)

Hence $|E_1'| \geq \sum_{i=1}^{n/2} |H_i| = n^2/(4D)$, as required. This is summarized in the following corollary.

COROLLARY 2.8. *For all integer numbers $n \geq 2$ and $1 \leq D \leq n - 1$, $f_S(D, n) = \bar{f}_S(D, n) \geq n^2/4D$.*

2.2.2. Directed graphs and distance-preserving labeling schemes. We next turn to proving a lower bound on $\bar{f}_S^{dir}(D, n) = f_S^{dir}(D, n)$.

Consider again the digraph $\vec{G}_1$ mentioned in section 2.2.1. Recall that the only subgraph D-preserver of $\vec{G}_1 = (V_1, \vec{E}_1)$ is the digraph itself (see inequality (2) and, also, any *undirected* Steiner D-preserver of this graph requires $\Omega(n^2/D)$ edges. However, as we mentioned, this digraph does admit a *directed* Steiner 1-preserver $G_1' = (V_1', E_1', \omega)$ with a *linear* number of edges. Specifically, $V_1' = V_1 \cup \{s_l, s_r\}$. Every vertex $x \in X \subseteq V_1$ is connected via an outgoing arc $\langle x, s_l \rangle$ to s_l, and via an incoming arc $\langle s_r, x \rangle$ to s_r. Also, every vertex $y \in Y \subseteq V_1$ is connected via an incoming arc $\langle s_l, y \rangle$ to s_l, and via an outgoing arc $\langle y, s_r \rangle$ to s_r. All these arcs are of weight $1/2$. The paths between y_i and z_i for $i = 1, 2, \ldots, n/2D$ are not modified. It is easy to see that for every pair of vertices $u, w \in V_1$, $d_{G_1'}(u, w) = d_{\vec{G}_1}(u, w)$. Also, $|E_1'| \leq 3/2n + n/D$. Hence, the digraph $\vec{G}_1$ cannot serve as an example that shows that $f_S^{dir}(D, n) = \Omega(n^2/D)$. Furthermore, we will show in section 2.3 that this claim is not true, and $f_S^{dir}(D, n) = O(\frac{n^2 \log D}{D \log n})$. In particular, it will follow that for $D = O(1)$, for any digraph there is a *directed Steiner* D-preserver with $O(n^2/\log n)$ arcs, where all the arcs are of weight 1 or 1/2. This separates the directed case from the undirected one, as $f_S(D, n) = \Omega(n^2/D)$ (see Corollary 2.8). Generalizing this upper bound, it will be shown there that for any digraph with $O(n^2/2^{\log^\gamma n})$ arcs, $0 < \gamma < 1$, a factor of $\Theta(\frac{\log^{1-\gamma} n}{\log \log n})$ can be "saved" using a diSteiner 1-preserver. Furthermore, some constant factor can be "saved" all the way to $n^{3/2}$. We next argue that there are n-vertex graphs G with $m = \Omega(n^{3/2})$ arcs such that any diSteiner 1-preserver of G contains at least m arcs.

Let $G = (U, W, E)$ be a bipartite graph with girth greater than 4. In other words, G contains no subgraph isomorphic to $K_{2,2}$ (the complete bipartite graph with two vertices in each bipartition).

We next argue that every diSteiner 1-preserver of G contains at least $|E|$ arcs.

Lemma 2.9. *Let $G' = (V', E', \omega)$ be a diSteiner 1-preserver of G. Then $|E'| \geq |E|$.*

Proof. Let G' be a diSteiner 1-preserver of the bipartite graph $G = (U, W, E)$. It follows that for any edge $e = (u, w) \in E$ there exists a path $P_e = P_{u,w}$ in G' of length 1. Associate such a path P_e with every edge $e \in E$ (if there are several such paths, pick one of them arbitrarily). We next argue that

$$\left| \bigcup_{e \in E} E'(P_e) \right| \geq |E|.$$

This would imply $|E'| \geq |E|$, as $|E'| \geq |\bigcup_{e\in E} E'(P_e)|$.

Consider an arbitrary ordering $(e_1, e_2, \ldots, e_{|E|})$ of the edges of E. Let $\mathcal{E}_k = \bigcup_{i=1}^k E'(P_{e_i})$.

Lemma 2.10. *$|\mathcal{E}_k| \geq k$, for every integer k, $1 \leq k \leq |E|$.*

Proof. The proof is by induction on k. For the induction base ($k = 1$), note that $|\mathcal{E}_1| = |E'(P_{e_1})| \geq 1$.

Assume the induction hypothesis for some k, $1 \leq k \leq |E| - 1$. It remains to argue that $|\mathcal{E}_{k+1} \setminus \mathcal{E}_k| \geq 1$. Let $e_{k+1} = (u, w)$. Let $\mathcal{E}(u, w) = \{(u', w') \in \{e_1, e_2, \ldots, e_k\} \mid E'(P_{u',w'}) \cap E'(P_{u,w}) \neq \emptyset\}$. Observe that for any edge $(u', w') \in \mathcal{E}(u, w)$, either $u = u'$ or $w = w'$. Indeed, otherwise let $s \in V'(P_{u,w}) \cap V'(P_{u',w'})$. Denote by $P_{u,s}$ (resp., $P_{u',s}$) the subsegment of $P_{u,w}$ (resp., $P_{u',w'}$) from u (resp., u') to s, and by $P_{s,w}$ (resp., $P_{s,w'}$) the subsegment of $P_{u,w}$ (resp., $P_{u',w'}$) from s to w (resp., w'). Note that $1 = |P_{u,w}| = |P_{u,s}| + |P_{s,w}| = |P_{u',w'}| = |P_{u',s}| + |P_{s,w'}|$. Suppose for contradiction that $|P_{u,s}| < |P_{u',s}|$. But then $d'_G(u, w') \leq |P_{u,s}| + |P_{s,w'}| < |P_{u',s}| + |P_{s,w'}| = |P_{u',w'}| = 1 \leq d_G(u, w')$, i.e., $d_{G'}(u, w') < d_G(u, w')$, contradiction. The assumption $|P_{u',s}| < |P_{u,s}|$ yields a contradiction in an analogous way. Hence, $|P_{u',s}| = |P_{u,s}|$.

It follows that $d_{G'}(u, w) = d_{G'}(u', w') = d_{G'}(u, w') = d_{G'}(u', w) = 1 = d_G(u, w) = d_G(u', w') = d_G(u, w') = d_G(u', w)$. That is, $(u, w), (u', w'), (u, w'), (u', w) \in E$, contradicting the assumption that no $K_{2,2}$ is contained in G.

So, for any edge $(u', w') \in \mathcal{E}(u, w)$ either $u = u'$ or $w = w'$. Note also that as $(u, w) \notin \{e_1, \ldots, e_k\}$, it follows that $(u, w) \notin \mathcal{E}(u, w)$, and thus either $u \neq u'$ or $w \neq w'$. Let $\mathcal{E}^u(u, w) = \{(u', w') \in \mathcal{E}(u, w) \mid u = u'\}$ and $\mathcal{E}^w(u, w) = \{(u', w') \in \mathcal{E}(u, w) \mid w = w'\}$; as we argued, $\mathcal{E}(u, w) = \mathcal{E}^u(u, w) \cup \mathcal{E}^w(u, w)$, and, $\mathcal{E}^u(u, w) \cap \mathcal{E}^w(u, w) = \emptyset$.

We next define an order relation $\leq_v$ of the vertices of $V'(P_{u,w})$ as follows. For a pair of vertices $x, y \in V'(P_{u,w})$, $x \leq_v y$ if and only if $d_{G'}(u, x) \leq d_{G'}(u, y)$.

Observe that for any edge $(u, w') \in \mathcal{E}^u(u, w)$, its corresponding path $P_{u,w'}$ "branches out" of the path $P_{u,w}$ at some point. Let $s(w')$ be the biggest vertex in $V'(P_{u,w}) \cap V'(P_{u,w'})$ with respect to the order relation $\leq_v$. We also define an order relation $\leq_e$ on the edges of $\mathcal{E}^u(u, w)$ as follows. For a pair of edges $(u, w_1), (u, w_2) \in \mathcal{E}^u(u, w)$, $(u, w_1) \leq_e (u, w_2)$ if and only if $s(w_1) \leq_v s(w_2)$.

Analogously, for any edge $(u', w) \in \mathcal{E}^w(u, w)$, let $s(u')$ be the smallest vertex of $V'(P_{u,w}) \cap V'(P_{u',w})$ with respect to the order relation $\leq_e$. The order relation $\leq_e$ on the edges of $\mathcal{E}^w(u, w)$ is defined in an analogous way.

Let (u, w') be the biggest edge in $\mathcal{E}^u(u, w)$, and (u', w) be the smallest edge in $\mathcal{E}^w(u, w)$ (both with respect to the order relation $\leq_e$; if there are several biggest edges, arbitrarily pick one of them).

Observe that by definition of $\mathcal{E}^u(u,w)$ and $\mathcal{E}^w(u,w)$, u,u',w,w' are distinct vertices of $V(G)$. Let $s(w')$ be the biggest vertex of $V'(P_{u,w}) \cap V'(P_{u,w'})$, and $s(u')$ be the smallest vertex of $V'(P_{u,w}) \cap V'(P_{u',w})$. It follows that $s(u') >_v s(w')$, as otherwise it would follow that the vertices u, u', w, and w' form $K_{2,2}$ in G, which is a contradiction. Let $P_{s(w'),s(u')}$ denote the subsegment of $P_{u,w}$ between $s(w')$ and $s(u')$. It remains to be argued that

$$E'(P_{s(w'),s(u')}) \cap \bigcup_{e \in \mathcal{E}_k} E'(P_e) = \emptyset. \tag{3}$$

Indeed, suppose for contradiction that there exists an edge $e \in \mathcal{E}_k$ such that $E'(P_e) \cap E'(P_{s(w'),s(u')}) \neq \emptyset$. It follows that $e \in \mathcal{E}(u,w) = \mathcal{E}^u(u,w) \cup \mathcal{E}^w(u,w)$. Recall that $\mathcal{E}^u(u,w) \cap \mathcal{E}^w(u,w) = \emptyset$. Hence $e \in \mathcal{E}^u(u,w)$ or $e \in \mathcal{E}^w(u,w)$.

Consider the case $e \in \mathcal{E}^u(u,w)$ (the case is $e \in \mathcal{E}^w(u,w)$ is analogous). Then $e = (u,w'')$ for some $w'' \in W$. Observe that as $E'(P_e) \cap E'(P_{s(w'),s(u')}) \neq \emptyset$, $s(u'), s(w') \in V'(P_e) \cap V'(P_{s(w'),s(u')})$, and so there exists a vertex $z \neq s(w')$ such that $z \in V'(P_e) \cap V'(P_{s(w'),s(u')})$. Note that $z \in V'(P_e)$ and $s(w') <_v z$. Observe also that $z \leq_v s(w'')$. It follows that $s(w') <_v s(w'')$, and so $(u,w') <_e (u,w'')$, contradicting the assumption that the edge (u,w') is the biggest in $\mathcal{E}^u(u,w)$ with respect to the order $\leq_e$. Now (3) follows. ◻

This completes the proof of Lemma 2.9. ◻

COROLLARY 2.11. *There are n-vertex graphs G with $m \geq (1/2 + o(1))n^{3/2}$ edges such that any diSteiner 1-preserver of G contains at least m arcs.*

Proof. It is well known (see, e.g., [8]) that there are bipartite graphs G_0 with $(1/2 + o(1))n^{3/2}$ edges with $girth(G_0) > 4$. The corollary follows by orienting all its arcs consistently from one bipartition to another, and using Lemma 2.9. ◻

In what follows we show that $\bar{f}_S^{dir}(D,n) = f_S^{dir}(D,n) = \Omega(\frac{n^2 \log D}{D \log n})$.

Let $\mathcal{G}$ be the family of graphs with a common vertex set V. The vertex set V is comprised of $X = \{x_1, x_2, \ldots, x_{n/2}\}$, $Y = \{y_1, y_2, \ldots, y_{n/(4D)}\}$, $Z = \{z_1, z_2, \ldots, z_{n/4D}\}$, and vertices of the paths connecting y_j to z_j for every $j = 1, 2, \ldots, n/4D$, $2D - 2$ vertices apart from y_j and z_j in each path. For every graph $G \in \mathcal{G}$, its edgeset contains the paths of length $2D - 1$ from y_j to z_j for every $j = 1, 2, \ldots, n/4D$. For every $j = 1, 2, \ldots, n/4D$ and $l = 1, 2, \ldots, 2D - 1$, let y_j^0 denote y_j, and y_j^l denote the vertex that is at distance l from y_j, and is located on the path connecting y_j and z_j. (In particular, $y_j^{2D-1} = z_j$.) In addition, for every $i = 1, 2, \ldots, n/2$, $j = 1, 2, \ldots, n/4D$, G contains precisely one arc from x_i to y_j^l, for some $l = 0, 1, \ldots, D - 1$. All the arcs are unit-weight. The family $\mathcal{G}$ consists of all the digraphs G that can be constructed this way.

It follows that

$$|\mathcal{G}| = D^{n/2 \cdot n/4D} = 2^{\frac{n^2 \log D}{8D}}. \tag{4}$$

We need the following definition.

DEFINITION 2.12. *The graph G' is a (D,g)-preserver of $G = (V,E)$ if for every pair of vertices $u, w \in V$ such that $d_G(u,w) \geq D$, $d_G(u,w) \leq d_{G'}(u,w) \leq d_G(u,w) + g$.*

LEMMA 2.13. *Let G_1' and G_2' be Steiner $(D, 1/3n)$-preservers of two distinct n-vertex graphs $G_1, G_2 \in \mathcal{G}$. Then $G_1' \neq G_2'$.*

Proof. As $G_1 \neq G_2$, there exists a pair $(i,j) \in \{1, 2, \ldots, n/2\} \times \{1, 2, \ldots, n/4D\}$ such that $\langle x_i, y_j^{l_1} \rangle \in E(G_1)$, $\langle x_i, y_j^{l_2} \rangle \in E(G_2)$, and $l_1 \neq l_2$. For these i and j,

$|d_{G_1}(x_i, z_j) - d_{G_2}(x_i, z_j)| \geq 1$. Observe also that as $l_1, l_2 \leq D-1$, it follows that $d_{G_1}(x_i, z_j), d_{G_2}(x_i, z_j) \geq (2D-1)-(D-1)+1 = D+1$. It follows that $|d_{G'_1}(x_i, z_j) - d_{G'_2}(x_i, z_j)| \geq |d_{G_1}(x_i, z_j) - d_{G_2}(x_i, z_j)| - 2/3n \geq 1 - 2/3n > 0$, for any $n = 1, 2, \ldots$. Hence, $d_{G'_1}(x_i, z_j) \neq d_{G'_2}(x_i, z_j)$. It follows that $G'_1 \neq G'_2$. □

Fix n and consider the family $\mathcal{G}$ of n-vertex digraphs discussed above. Let $V = (v_1, v_2, \ldots, v_n)$ be an arbitrary ordering of the (common to all graphs of $\mathcal{G}$) vertex set V. For a distance labeling scheme $(\mathcal{M}, \mathcal{D})$ and a graph $G \in \mathcal{G}$, let $\mathcal{M}(G) = \mathcal{M}(G, v_1) \cdot \mathcal{M}(G, v_2) \cdot \cdots \cdot \mathcal{M}(G, v_n)$, where "$\cdot$" stands for concatenation. It is easy to see that, without loss of generality, we can restrict our attention to labeling schemes that assign labels of the same size to all the vertices for all graphs of $\mathcal{G}$.

Lemma 2.14. *Let $(\mathcal{M}, \mathcal{D})$ be a distance-labelling D-preserving scheme and $G_1, G_2 \in \mathcal{G}$, $G_1 \neq G_2$. Then $\mathcal{M}(G_1) \neq \mathcal{M}(G_2)$.*

Proof. Similar to the proof of Lemma 2.13, since $G_1 \neq G_2$, there exists a pair of vertices $x_i, z_j \in V$ such that $d_{G_1}(x_i, z_j), d_{G_2}(x_i, z_j) \geq D$, and $d_{G_1}(x_i, z_j) \neq d_{G_2}(x_i, z_j)$.

As $(\mathcal{M}, \mathcal{D})$ is a D-preserving scheme, it follows that $\mathcal{D}(\mathcal{M}(G_1, x_i), \mathcal{M}(G_1, z_j)) = d_{G_1}(x_i, z_j)$ and $\mathcal{D}(\mathcal{M}(G_2, x_i), \mathcal{M}(G_2, z_j)) = d_{G_2}(x_i, z_j)$. Hence, $\mathcal{D}(\mathcal{M}(G_1, x_i), \mathcal{M}(G_1, z_j)) \neq \mathcal{D}(\mathcal{M}(G_2, x_i), \mathcal{M}(G_2, z_j))$. Hence, either $\mathcal{M}(G_1, x_i) \neq \mathcal{M}(G_2, x_i)$ or $\mathcal{M}(G_1, z_j) \neq \mathcal{M}(G_2, z_j)$ (or both). In either case, $\mathcal{M}(G_1) \neq \mathcal{M}(G_2)$. □

Let φ be an arbitrary representation function of the Steiner $(D, 1/3n)$-preservers of graphs from the family $\mathcal{G}$. Specifically, with each graph $G \in \mathcal{G}$, φ associates a bit string of fixed length k, that determines uniquely some specific Steiner $(D, 1/3n)$-preserver G' of G. Note that by Lemma 2.13, φ is injective. Indeed, if $G' = \varphi(G_1) = \varphi(G_2)$, then G' is a Steiner $(D, 1/3n)$-preserver of both G_1 and G_2, and so, by Lemma 2.13, $G_1 = G_2$. Hence, by (4), we have the following corollary.

Corollary 2.15. *For every representation function φ of the Steiner $(D, 1/3n)$-preservers of $\mathcal{G}$, the length k (in terms of the number of bits) of the representation bit string is $k \geq \log|\{\varphi(G) \mid G \in \mathcal{G}\}| = \log|\mathcal{G}| = \frac{n^2 \log D}{8D}$.*

Note that if the representation function is allowed to use representations of different lengths, then the number of graphs that can be encoded by it is greater than the one in Corollary 2.15 by at most a constant factor.

Analogously, Lemma 2.14 implies a lower bound on D-preserving distance labeling schemes. Note that all the lower bounds in this section apply both to the directed and undirected graphs. However, for undirected Steiner graphs stronger lower bounds were shown in section 2.2.1. This is not the case for the distance labeling schemes, where the lower bound below is the strongest that we are able to prove.

Corollary 2.16. *Every distance labeling D-preserving scheme requires labels of size $\Omega(\frac{n \log D}{D})$ bits.*

Intuitively, the last stage of the proof of the lower bound $f_S^{dir}(D, n) = \Omega(\frac{n^2 \log D}{D \log n})$ is proving that using nonrational (or even rational but having a very large denominator) weights cannot help saving arcs of the diSteiner D-preservers. This is done in the next theorem. The technique of getting rid of the nonrational weights in a Steiner graph that is used in the proof is adapted from [1], where Steiner spanners with a multiplicative approximation of distances are studied.

Theorem 2.17. *For an integer $n \geq 2$, the family of n-vertex digraphs $\mathcal{G}$ defined above, and an integer D, $1 \leq D \leq n-1$, let $\rho : \mathcal{G} \to \mathcal{G}'$ be a function assigning to every digraph $G \in \mathcal{G}$ a diSteiner D-preserver G'. Then there exists a digraph $G \in \mathcal{G}$ such that $G' = \rho(G)$ contains $\Omega(\frac{n^2 \log D}{D \log n})$ arcs.*

Proof. Consider a mapping $\rho' : \mathcal{G}' \to \mathcal{G}''$ which, given a digraph $G' = (V', E', \omega)$, constructs a digraph $G'' = (V', E', \omega')$, where for every arc $e \in E'$, $\omega'(e)$ is defined to be the smallest rational number with denominator $1/3n^3$ that is no smaller than $\omega(e)$. Let $\rho'' : \mathcal{G} \to \mathcal{G}''$ be the composition of ρ and ρ'.

Suppose for contradiction that for any digraph $G \in \mathcal{G}$, its diSteiner D-preserver $G' = \rho(G)$ contains less than $\frac{n^2 \log D}{6 \cdot (8D \log n)}$ arcs. In particular, it follows that for any digraph $G \in \mathcal{G}$, its diSteiner D-preserver $G' = (V', E', \omega)$ has at most n^2 vertices. Hence for any pair of vertices $u, w \in V'$, any simple path from u to w in G' contains no more than n^2 arcs. As for every arc $e \in E'$, $|\omega(e) - \omega'(e)| \leq 1/3n^3$, it follows that for any simple path P from u to w in G', $|\omega(P) - \omega(P')| \leq n^2/3n^3 = 1/3n$.

As G' is a diSteiner D-preserver of G, it follows that $\rho'(G') = G''$ is a diSteiner $(D, 1/3n)$-preserver of G. Observe also that for any $G \in \mathcal{G}$, the digraphs $G' = \rho(G)$ and $G'' = \rho''(G)$ have the same arcset. By our assumption, for every digraph $G \in \mathcal{G}$, $G' = \rho(G)$ contains less than $\frac{n^2 \log D}{6 \cdot (8D \log n)}$ arcs. It follows that for every digraph $G \in \mathcal{G}$, $G'' = \rho''(G)$ contains less than $\frac{n^2 \log D}{6 \cdot (8D \log n)}$ arcs. Observe also that for any arc $e \in E(G'')$, its weight in G'' is a rational number. As all the distances in G are no greater than $n - 1$ and G'' is a diSteiner $(D, 1/3n)$-preserver, we assume, without loss of generality, that all the arcs in G'' have weight that is no greater than n. Hence, every arc $e \in E(G'')$ can be represented by a bit string $\alpha(e)$ of length $6 \log n$, by writing down the identities of its endpoints ($2 \log n$ bits) and the numerator of its weight (at most $\log n^4 = 4 \log n$ bits).

The representation function φ is now formed out of ρ by concatenating in an arbitrary but fixed order the strings $\alpha(e)$ for different arcs $e \in E(G'')$. Observe that for any digraph $G \in \mathcal{G}$, $\varphi(G)$ determines uniquely a diSteiner $(D, 1/3n)$-preserver G'' of G and $\varphi(G)$ contains $\frac{n^2 \log D}{6 \cdot (8D \log n)} 6 \log n = \frac{n^2 \log D}{8D \log n}$ bits. However, this contradicts Corollary 2.15.

Hence there is a digraph $G \in \mathcal{G}$ such that its diSteiner D-preserver $\rho(G) = G'$ contains at least $\frac{n^2 \log D}{48D \log n}$ arcs. ∎

2.2.3. (D, g)-preservers.

In this section we prove a lower bound on the cardinality of *subgraph* (D, g)-preservers.

To facilitate the discussion about (D, g)-preservers, we generalize Definition 2.3 in the following way.

DEFINITION 2.18. *For $n \geq 2$, and $1 \leq D, g \leq n - 1$, let $f(D, g, n)$ be the minimal number such that for any n-vertex graph there exists a (D, g)-preserver with at most $f(D, g, n)$ edges, and let $\bar{f}(D, g, n)$ be the maximal number of edges in an n-vertex graph whose only subgraph (D, g)-preserver is the graph itself.*

The following lemma follows directly from the definition.

LEMMA 2.19. *For $n \geq 2$ and $1 \leq D, g \leq n - 1$, we have $f(D, g, n) \geq \bar{f}(D, g, n)$.*

However, unlike the case with no additive error, no upper bound on $f(D, g, n)$ in terms of $\bar{f}(D, g, n)$ is known to the authors.

We next show a lower bound on $\bar{f}(D, g, n)$, which serves, consequently, as a lower bound on $f(D, g, n)$.

The lower bound on the size of an extremal n-vertex graph of girth g stands currently on $\Omega(n^{1+c_0/(g-1)})$ [5], for $c_0 = 4/3$; Erdős conjectured that $c_0 = 2$.

THEOREM 2.20. *For integer numbers n, D and g, $D, g \geq 2$, and n sufficiently large, $f(D, g, n) \geq \bar{f}(D, g, n) \geq \frac{n^{1+c_0/(g+2)}}{2g \cdot D^{c_0/(g+2)}}$.*

Proof. Set $L = \lfloor n/2D \rfloor$. There exists a constant $1 \leq c_0 \leq 2$ such that there exists an L-vertex graph $G_0 = (V_0, E_0)$ with $girth(G_0) \geq g + 2$ and $|E_0| \geq L^{1+c_0/(g+2)}$ (cf.

[8]). Denote the vertices of G_0 by the numbers $1, 2, \ldots, L$ (that is, $V_0 = \{1, 2, \ldots, L\}$).

To build the graph $G^{(D,g)}$, we begin with L paths of length D: vertices v_{ij}, $i = 1, 2, \ldots, L$, $j = 1, 2, \ldots, D$, and edges $(v_{ij}, v_{i,j+1})$, $i = 1, 2, \ldots, L$, $j = 1, 2, \ldots, D-1$.

Add $L \cdot D/\lfloor g/2 \rfloor$ vertices w_{ij}, $i = 1, 2, \ldots, L$, $j = 1, 2, \ldots, D/\lfloor g/2 \rfloor$, and for any $i = 1, 2, \ldots, L$, $j = 1, 2, \ldots, D/\lfloor g/2 \rfloor$ connect v_{i1} to w_{ij} by a path of length $\lfloor g/2 \rfloor$. (Small corrections should be made if $\lfloor g/2 \rfloor$ does not divide D.)

For each j, $j = 1, 2, \ldots, D/\lfloor g/2 \rfloor$, construct an isomorphic copy of G_0 using the vertices $\{w_{ij}\}_{i=1}^{L}$. Specifically, for each j, $j = 1, 2, \ldots, D/\lfloor g/2 \rfloor$, for every $i, h = 1, 2, \ldots, L$, add the edge (w_{ij}, w_{hj}) if and only if $(i, h) \in E_0$.

The number of vertices is $L \cdot (D + \lfloor g/2 \rfloor \cdot D/\lfloor g/2 \rfloor) = 2LD = 2D\lfloor n/(2D) \rfloor \leq n$; add $n - 2DL$ vertices to one of the paths to absorb the slack, giving $G^{(D,g)}$ exactly n vertices (i.e., $|V^{(D,g)}| = n$).

The number of the edges is

$$\begin{aligned} |E^{(D,g)}| &\geq L \cdot (D-1) + (L \cdot D/\lfloor g/2 \rfloor) \cdot \lfloor g/2 \rfloor + n - 2DL \\ &\quad + \lfloor n/2D \rfloor^{1+c_0/(g+2)} \cdot D/\lfloor g/2 \rfloor) \\ &\geq n - \lfloor n/2D \rfloor + \frac{n^{1+c_0/(g+2)} \cdot 2D}{2^{1+c_0/(g+2)} D^{1+c_0/(g+2)} \lfloor g/2 \rfloor} \;\geq\; \frac{n^{1+c_0/(g+2)}}{2gD^{c_0/(g+2)}} \,. \end{aligned}$$

Let us argue that $G^{(D,g)}$ is the only subgraph (D, g)-preserver of itself.

Indeed, removing a path edge $e = (v_{ij}, v_{i+1,j})$ for some $i = 1, 2, \ldots, D-1$, $j = 1, 2, \ldots, L$ makes the graph disconnected, and, in particular, $d_G(w_{ij'}, v_{iD}) \geq D$, and $d_{G_e}(w_{ij'}, v_{iD}) = \infty$, for any $j' = 1, 2, \ldots, D/\lfloor g/2 \rfloor$.

Removing an edge from a path that connects v_{i1} with w_{ij} for some $i = 1, 2, \ldots, L$, $j = 1, 2, \ldots, D/\lfloor g/2 \rfloor$ increases the distance between w_{ij} and v_{iD} by at least $g+1$.

Finally, removing an edge (w_{ij}, w_{hj}) increases the distance from w_{hj} to v_{iD} by at least $g+1$, since for any $j = 1, 2, \ldots, D/\lfloor g/2 \rfloor$, the graph $G^{(D,g)}(\{w_{ij} \mid i = 1, 2, \ldots, L\})$ has girth equal to $g+2$. □

2.3. Upper bounds. We start by introducing some definitions.

Definition 2.21. *Given a path $P = (v_0, v_1, \ldots, v_s)$, and an arc $\langle v_i, v_{i+1} \rangle \in P$, let prefix$(P, \langle v_i, v_{i+1} \rangle)$ (resp., suffix$(P, \langle v_i, v_{i+1} \rangle)$) denote the path $(v_0, \ldots, v_i)$ (resp., $(v_{i+1}, \ldots, v_s)$). The* head *(resp.,* tail*) of P, denoted head(P) (resp., tail(P)) is v_0 (resp., v_s).*

For a digraph (resp., undirected graph) $G = (V, E)$ and an arc (resp., edge) $e \in E$, let G_e denote the digraph (resp., undirected graph) $(V, E \setminus \{e\})$.

In an undirected graph $G = (V, E)$, given a walk $P = (v_0, \ldots, v_s)$, and an edge $e = (v_i, v_{i+1}) \in P$, the (e, v_i)-endpoint *of P, denoted endpoint(P, e, v_i), is v_0. The* (e, v_i)-subpath *of P, denoted by subpath(P, e, v_i), is $(v_0, \ldots, v_i)$.*

Given two walks $P_1 = (v_0, \ldots, v_s)$ and $P_2 = (v_s, \ldots, v_{t+s})$, $t, s \geq 0$, the concatenation *$P_1 \cdot P_2$ is the walk $(v_0, \ldots, v_{t+s})$. Obviously, the concatenation is associative, and so $P_1 \cdot P_2 \cdot \cdots \cdot P_r$ is well defined, whenever for any $i = 1, \ldots, r-1$, $P_i \cdot P_{i+1}$ is defined.*

Definition 2.22. *Given a digraph $G = (V, E)$ and a positive integer distance threshold D, the* D-path associated with an arc *e, denoted by $P(e, D)$, is one of the shortest paths between its endpoints head$(P(e, D))$ and tail$(P(e, D))$ such that*

$$d_G(head(P(e,D)), tail(P(e,D))) = |P(e,D)| \geq D, \tag{5}$$

$$d_{G_e}(head(P(e,D)), tail(P(e,D))) > d_G(head(P(e,D)), tail(P(e,D))). \tag{6}$$

Given an undirected graph $G = (V, E)$ and a positive integer D, the D-path *associated with the edge $e = (v, z)$, denoted $P(e, D)$, is one of the shortest paths between $endpoint(P(e, D), e, v)$ and $endpoint(P(e, D), e, z)$ such that*

$$\begin{aligned} & d_G(endpoint(P(e,D),e,v), endpoint(P(e,D),e,z)) \ = \ |P(e,D)| \geq D, \\ & d_{G_e}(endpoint(P(e,D),e,v), endpoint(P(e,D),e,z)) \ > \\ & d_G(endpoint(P(e,D),e,v), endpoint(P(e,D),e,z)). \end{aligned}$$

Note that such a path may not exist, and, on the other hand, there may be several such paths. In the latter case, set $P(e, D)$ to be such an arbitrary path.

Throughout the section, whenever the value of D is clear from the context, we use the notation $P(e)$ instead of $P(e, D)$.

2.3.1. Distance preservers. We start by presenting an almost matching (up to a constant factor of 4) upper bound on the size of possible distance D-preservers.

LEMMA 2.23. *For integer numbers $n \geq 2$ and $1 \leq D \leq n - 1$,*

$$f^{dir}(D, n) = \bar{f}^{dir}(D, n) \leq 2n(n-1)/(D+1), \quad f(D, n) = \bar{f}(D, n) \leq n(n-1)/(D+1).$$

Proof. We first consider the directed case. Suppose, without loss of generality, that for every arc $e \in E$, the path $P(e) = P(e, D)$ exists. (Indeed, an arc e for which $P(e)$ does not exist can be safely removed from the graph.)

Consider some vertex $v \in V$. We next argue that for any two arcs that are outgoing from v, $e_1 = \langle v, z_1 \rangle$, $e_2 = \langle v, z_2 \rangle$,

$$V(suffix(P(e_1), e_1)) \cap V(suffix(P(e_2), e_2)) = \emptyset.$$

Suppose for contradiction that some vertex $w \in V(suffix(P(e_1), e_1)) \cap V(suffix(P(e_2), e_2))$. Then, $d_{G_{e_1}}(head(P(e_1)), tail(P(e_1))) > d_G(head(P(e_1)), tail(P(e_1)))$, and $d_{G_{e_2}}(head(P(e_2)), tail(P(e_2))) > d_G(head(P(e_2)), tail(P(e_2)))$.

For $i = 1, 2$, let P'_i, P''_i and P'''_i be the segments of $P(e_i)$ from $head(P(e_i))$ to v, from v to w, and from w to $tail(P(e_i))$, respectively; see Figure 2.

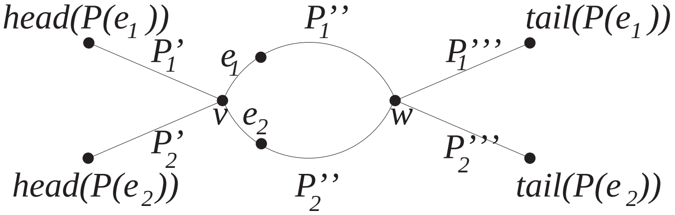


FIG. 2. *The subpaths of $P(e_1)$ and $P(e_2)$.*

Note that since $P(e_1)$ is the shortest path between $head(P(e_1))$ and $tail(P(e_1))$ in G, $d_G(head(P(e_1)), tail(P(e_1))) = |P'_1| + |P''_1| + |P'''_1|$.

Consider the walk $P_{12} = P'_1 \cdot P''_2 \cdot P'''_1$. Note that P_{12} is a walk between $head(P_1)$ and $tail(P_1)$ in $E \setminus \{e_1\}$. Hence,

$$\begin{aligned} |P'_1| + |P''_2| + |P'''_1| &\geq d_{G_{e_1}}(head(P(e_1)), tail(P(e_1))) \\ &> d_G(head(P(e_1)), tail(P(e_1))) \ = \ |P'_1| + |P''_1| + |P'''_1|. \end{aligned}$$

Hence $|P''_2| > |P''_1|$. However, analogously, it follows that $|P''_1| > |P''_2|$—a contradiction.

Therefore, the set $\mathcal{P}_{out}(v) = \{\mathit{suffix}(P(\langle v,z\rangle), \langle v,z\rangle) \mid \langle v,z\rangle \in E\}$ consists of vertex-disjoint paths.

Analogously, it follows that the set $\mathcal{P}_{in}(v) = \{\mathit{prefix}(P(\langle z,v\rangle), \langle z,v\rangle) \mid \langle z,v\rangle \in E\}$ consists of vertex-disjoint paths.

Note that for every vertex $v \in V$ and path $P \in \mathcal{P}_{in}(v) \cup \mathcal{P}_{out}(v)$, the node v does not belong to $V(P)$. Thus,

$$\sum_{P \in \mathcal{P}_{out}(v)} |V(P)|, \sum_{P \in \mathcal{P}_{in}(v)} |V(P)| \leq |V \setminus \{v\}| = n - 1.$$

Thus,

$$\sum_{v \in V} \left(\sum_{P \in \mathcal{P}_{in}(v)} |V(P)| + \sum_{P \in \mathcal{P}_{out}(v)} |V(P)| \right) \leq 2n(n-1).$$

Also, since for any arc $e \in E$, $|P(e)| \geq D$,

$$\begin{aligned}
&\sum_{v \in V} \left(\sum_{P \in \mathcal{P}_{in}(v)} |V(P)| + \sum_{P \in \mathcal{P}_{out}(v)} |V(P)| \right) \\
&= \sum_{v \in V} \left(\sum_{\langle z,v\rangle \in E} |V(\mathit{prefix}(P(\langle z,v\rangle), \langle z,v\rangle))| + \sum_{\langle v,z\rangle \in E} |V(\mathit{suffix}(P(\langle v,z\rangle), \langle v,z\rangle))| \right) \\
&= \sum_{\langle v,z\rangle \in E} (|V(\mathit{prefix}(P(\langle v,z\rangle), \langle v,z\rangle))| + |V(\mathit{suffix}(P(\langle v,z\rangle), \langle v,z\rangle))|) \\
&= \sum_{\langle v,z\rangle \in E} (|\mathit{prefix}(P(\langle v,z\rangle), \langle v,z\rangle)| + 1 + |\mathit{suffix}(P(\langle v,z\rangle), \langle v,z\rangle)| + 1) \\
&= \sum_{\langle v,z\rangle \in E} (|P(\langle v,z\rangle)| + 1) \;=\; \sum_{\langle v,z\rangle \in E} |P(\langle v,z\rangle)| + |E| \;\geq\; |E| \cdot D + |E|.
\end{aligned}$$

Thus, $|E| \cdot (D+1) \leq 2n(n-1)$.

For an undirected graph $G = (V,E)$, the analogous argument provides an upper bound which is smaller by a factor of 2. □

Note that the inequalities in Lemma 2.23 are tight for $D = 1$, since there is a graph (n-vertex clique K_n) with $n \cdot (n-1)/(D+1) = n \cdot (n-1)/2$ edges, in which removal of any edge results in increasing the distance between some pair of vertices that are already at a distance of at least $D = 1$. Also, there is a digraph (complete n-vertex digraph) with $2n \cdot (n-1)/(D+1) = n \cdot (n-1)$ arcs, with the same property.

The next theorem indicates that the product $D \cdot f(D,n)$ is independent of D and equal to $\Theta(n^2)$.

THEOREM 2.24 (distance×size preservation). *For* $n = 2,3,\ldots,$ *and* $D = 1,2,\ldots,n-1$,

$$n^2/4D \;\leq\; f_S(D,n) \;\leq\; f(D,n) \;\leq\; n(n-1)/(D+1), \tag{7}$$

$$n^2/2D \;\leq\; f^{dir}(D,n) \;\leq\; 2n(n-1)/(D+1). \tag{8}$$

Proof. Both upper bounds follow from Lemma 2.23. The lower bound of inequality (7) follows from Theorem 2.8. The lower bound of inequality (8) follows from (2); $f^{dir}(D,n) = \bar{f}^{dir}(D,n) \geq n^2/2D$. □

We next prove a tight up to a constant factor upper bound on $f_S^{dir}(D,n)$.

Consider an n-vertex digraph $G=(V,E)$ with $m=\Omega(n^{3/2})$ arcs. Suppose $V=\{1,2,\ldots,n\}$. The digraph G can be represented by its $n\times n$ adjacency matrix $M(G)$, whose entry (i,j) is 1 if and only if $\langle i,j\rangle\in E$, and 0 otherwise. Suppose, without loss of generality, that the digraph contains no loops (that is, arcs $\langle i,i\rangle$ for some $i\in V$) as the latter can be removed from the digraph with no affect on the distances. Set $c''=1+\nu_1$ for some arbitrarily small positive constant $\nu_1>0$. Denote $p=m/(c''n^2)$.

LEMMA 2.25. *$M(G)$ contains an $a\times a$ submatrix containing all 1's with $a=\lfloor c'\log n/\log(1/p)\rfloor$, for $c'=1-\nu_2$ for some arbitrarily small positive constant $\nu_2>0$.*

Remark. Such a matrix corresponds to $K_{a,a}$, that is, complete bipartite subgraph of size $a\times a$ with all arcs oriented consistently from one bipartition of the subgraph to another.

Proof. Following Zarankiewicz, let us denote by $k_a(n)$ the least number m such that any n-vertex digraph G with at least m arcs contains a $K_{a,a}$. The assertion of the lemma is a corollary of the following result from [18, chapter 5].

THEOREM 2.26 (see [18]). *If $n\binom{m/n}{a}\ge(a-1)\binom{n}{a}$, then $k_a(n)\le m$.*

To show that the assumption of Theorem 2.26 is satisfied, it is enough to argue that

$$n\cdot\left(\frac{m/n}{n}\cdot\frac{m/n-1}{n-1}\cdot\ldots\cdot\frac{m/n-a+1}{n-a+1}\right)\ \ge\ a.$$

As $m/n=\Omega(\sqrt{n})$ and $a=O(\log n)$, it follows that for any sufficiently large n and any $i=1,2,\ldots,a-1$,

$$\frac{m/n-i}{n-i}\ge\frac{m/(c''n)}{n}.$$

Hence, it is sufficient to argue that $n(m/(c''n))^a=n\cdot p^a\ge a$. Substituting $a=c'\log n/\log(1/p)$ implies $n^{1-c'}\ge a$, and the latter is true for sufficiently large n (as $a=O(\log n)$). Theorem 2.26 implies $k_a(n)\le m$. The assertion of the lemma now follows from the definition of $k_a(n)$. ☐

Let $m_0=m=n^2/D$ for some D be the number of arcs in $G_0=G$, and $p_0=p=m_0/(c''n^2)$ be the "density" of the arcs. Set $\epsilon=\frac{\log D}{\log n}$ (i.e., $D=2^{\epsilon\log n}$). Set also $S_0=0$ to be the number of arcs inserted into the diSteiner graph so far. By Lemma 2.25, G contains a subgraph isomorphic to K_{a_0,a_0} with $a_0=c'\log n/(\log 1/p_0)$. Pick such a subgraph and represent it with a diSteiner vertex s (in addition to $2a_0$ original vertices) and $2a_0$ appropriately oriented arcs of weight $1/2$ each connecting s with the original vertices. The orientation of these arcs is the following: all the arcs between s and "left-hand" vertices (those that had only outgoing arcs in the chosen subgraph) are incoming into s, and all the other arcs are outgoing from s. The constructed structure is inserted into the diSteiner graph, and the charge S is updated from $S_0=0$ to $S_1=S_0+2a_0=2a_0$. Delete the arcs of chosen subgraph from G_0, and denote the obtained digraph G_1. The density p changes according to $p_1=p_0-a_0^2/(c''n^2)$. If the number of arcs in G_1 is still greater than $\mu\cdot\frac{n^2(\log D+\log e)}{D\cdot\log n}$ for some arbitrarily small constant $\mu>0$, repeat this procedure with $a_1=c'\log n/(\log 1/p_1)$. Observe that the condition on the number of arcs implies that $a_1\ge1$, and so in a finite number r of iterations we are left with a digraph G_r with at most $\mu\cdot\frac{n^2(\log D+\log e)}{D\cdot\log n}$ arcs. When the number of arcs left is at most $\mu\cdot\frac{n^2(\log D+\log e)}{D\cdot\log n}$, these arcs are inserted into the diSteiner graph G'.

LEMMA 2.27. *The constructed digraph G' is a diSteiner 1-preserver of G.*

Proof. Consider some arc $\langle u, w\rangle \in E$. At either one of the iterations e was replaced by two arcs $\langle u, s\rangle, \langle s, w\rangle$ of weight $1/2$ each, for some new vertex s, or the arc e was inserted into G'. In either case $d_{G'}(u, w) = d_G(u, w) = 1$. It follows that for any pair of vertices $x, y \in V(G)$, $d_{G'}(x, y) \le d_G(x, y)$.

Also, it can be shown by induction on r that for any $x, y \in V(G)$, $d_G(x, y) \le d_{G'}(x, y)$. Intuitively, this is because whenever an isomorphic to a $K_{a,a}$ between $x_1, x_2, \ldots, x_a$ and $y_1, y_2, \ldots, y_a$ is replaced by a star of arcs $\langle x_1, s\rangle, \langle x_2, s\rangle, \ldots, \langle x_a, s\rangle$, $\langle s, y_1\rangle, \langle s, y_2\rangle, \ldots, \langle s, y_a\rangle$, no paths between x_i and x_j or y_i and y_j are formed. This is unlike the undirected case, where such a replacement could cause $d_{G'}(x_i, x_j) < d_G(x_i, x_j)$. This is, however, quite natural, as in the undirected case there are graphs for which any Steiner 1-preserver contains $\Omega(n^2)$ edges (see Theorem 2.24, inequality (7)).

It follows that G' is a diSteiner 1-preserver of G. □

Next, we calculate the number of arcs in G'.

LEMMA 2.28. *If n is sufficiently large, then*

$$S_r \;\le\; \frac{2c''}{c' - \epsilon} \cdot \frac{n^2}{D} \cdot \frac{\log D + \log e}{\log n}, \tag{9}$$

where $\epsilon = \frac{\log D}{\log N}$.

Proof. Observe that $S_r = S_0 + 2\sum_{i=0}^{r-1} a_i = 2\sum_{i=0}^{r-1} a_i$. Denote $\Delta p_i = p_{i+1} - p_i$ for $i = 0, 1, \ldots, r-1$. Note that $\Delta p_i > 0$ for $i = 0, 1, \ldots, r-1$. Then $S_r/2 = \sum_{i=0}^{r-1} \frac{a_i}{\Delta p_i}\Delta p_i$. Observe that $\Delta p_i = p_{i+1} - p_i = \frac{a_i^2}{c'' n^2}$. Hence $\frac{a_i}{\Delta p_i} = c'' n^2/a_i$. By Lemma 2.25, $a_i \ge c'(\frac{\log n}{\log 1/p_i} - 1/c')$. Substituting $p_i \ge \mu \frac{\log D}{c'' D \log n}$ and $D = 2^{\epsilon \log n}$ implies that $\log 1/p_i \le \epsilon \log n - \log \mu\epsilon$. Hence $\frac{\log n}{\log 1/p_i} - 1/c' \ge (1 - \epsilon/c')\frac{\log n}{\log 1/p_i}$. Therefore, $a_i/\Delta p_i \le \frac{c''}{c'}\frac{1}{1-\epsilon/c'}\frac{n^2 \log 1/p_i}{\log n} = \frac{c''}{c'-\epsilon} \cdot \frac{n^2 \log 1/p_i}{\log n}$. Hence

$$S_r/2 \le c'' \cdot \frac{1}{c' - \epsilon} n^2/\log n \sum_{i=0}^{r-1} \log 1/p_i \Delta p_i. \tag{10}$$

Observe that as $p_0 > p_1 > \ldots p_{r-1} > p_r > 0$, it follows that $\sum_{i=0}^{r-1} \log 1/p_i \Delta p_i$ is a Riemann sum of $\int_0^{p_0} (\log 1/p)dp$. Furthermore, $\Delta p_i = a_i^2/(2n^2) \le \frac{\log^2 n}{n^2}$. Hence Δp_i tends to 0 when n grows, for any $i = 0, 1, 2, \ldots, r-1$. Hence for any $\delta > 0$ there exists a sufficiently large n such that

$$\sum_{i=0}^{r-1} \log 1/p_i \Delta p_i \;\le\; \int_0^{p_0} (\log 1/p)dp + \delta \;\le\; p_0(\log 1/p_0 + 1) + \delta.$$

Now, the lemma follows from (10). □

COROLLARY 2.29. *For every n-vertex (di)graph with m edges (resp., arcs) the following statements hold.*

1. *There exists a diSteiner 1-preserver with $O(n^2/\log n)$ arcs.*
2. *If $m \le n^2/\log^c n$ for some $c > 0$, then there exists a diSteiner 1-preserver with $O(\frac{c \cdot n^2 \log\log n}{\log^{c+1} n})$ arcs.*
3. *If $m \le n^{1+\alpha}$, $0 < \alpha < 1$, then there exists a diSteiner 1-preserver with at most $\frac{2+\mu}{\alpha}(1-\alpha) \cdot m$ arcs for any arbitrarily small constant μ.*

4. *There exists a diSteiner D-preserver with $O(\frac{n^2 \log D}{D \log n})$ arcs. In other words,*

$$\bar{f}_S^{dir}(D,n) = f_S^{dir}(D,n) = \Theta\left(\frac{n^2 \log D}{D \log n}\right).$$

The weights of arcs in the aforementioned diSteiner graphs may be restricted to be either 1 *or* 1/2.

Proof. For assertion (1), use Lemma 2.28 with $\epsilon = 0$ (observe that the right-hand side of (9) is minimized by setting $\epsilon = 0$). It follows that $S_r \leq (2+\nu)n^2/\log n$, for some arbitrarily small constant $\nu > 0$. The assertion follows as the number of arcs in the diSteiner 1-preserver is

$$S_r + \mu \cdot \frac{n^2 \cdot \log D}{D \cdot \log n} = (2+\mu)n^2/\log n$$

for an arbitrarily small constant $\mu > 0$.

The assertion (2) follows analogously, noting that $\epsilon = \frac{\log D}{\log N} = c \log\log n/\log n$.

For assertion (3), note that $D = n^2/n^{1+\alpha} = n^{1-\alpha}$. That is, $\epsilon = 1-\alpha$. Now the assertion follows from Lemma 2.28.

For assertion (4), recall that by Theorem 2.24, for any n-vertex (di)graph there exists a subgraph D-preserver with $O(n^2/D)$ edges (resp., arcs). If $D = \Omega(n^\epsilon)$ for some constant $\epsilon > 0$, then $O(n^2/D) = O(\frac{n^2 \log D}{D \log n})$. Otherwise, if $D = 2^{\epsilon(n)\cdot \log n}$ for some $\epsilon(n)$ such that $\lim_{n\to\infty} \epsilon(n) = 0$, then the assertion follows from Lemma 2.28, and from the observation that a 1-preserver of a D-preserver of a graph G is a D-preserver of G.

Finally, the lower bound

$$\bar{f}_S^{dir}(D,n) = f_S^{dir}(D,n) = \Omega\left(\frac{n^2 \log D}{D \log n}\right)$$

follows from Theorem 2.17. □

Note that by Corollary 2.29, for any graph with at least $m = n^{5/3+\delta}$ edges (for any $\delta > 0$) there exists a diSteiner 1-preserver with strictly less than m arcs. This statement can be generalized to $m \geq c \cdot n^{3/2}$, for some small constant $c > 1$, by extracting subgraphs isomorphic to $K_{s,2}$ for different decreasing values of s whenever no $K_{3,3}$ can be extracted; see the discussion that follows Corollary 2.33. Note that the latter cannot be generalized much further, as by Corollary 2.11 there exist n-vertex graphs with $m = (1/2+o(1))n^{3/2}$ edges for which any diSteiner 1-preserver contains at least m arcs.

2.3.2. Algorithmic aspects. In this section we address some algorithmic aspects of our results concerning distance D-preservers. In particular, we devise a distance labeling D-preserving scheme with labels of size $O((n^2/D) \cdot \log^2 n)$. Recall that by Corollary 2.16 labels of size $O((n^2/D) \cdot \log D)$ are required.

THEOREM 2.30. *For integer numbers $n \geq 2$, $1 \leq D \leq n-1$, and an n-vertex graph (resp., digraph) with m edges (resp., arcs), there exists a constructible in $O(m^2 n)$ time subgraph D-preserver with at most $n(n-1)/(D+1)$ edges (resp., $2n(n-1)/(D+1)$ arcs).*

Proof. We prove the assertion for a digraph G; the proof of the slightly stronger statement for the undirected graphs is analogous.

The proof is, by induction, on the number of arcs in G, $|E| = m$. The induction base is $|E| \le \frac{2n\cdot(n-1)}{D+1}$. In this case $G' = (V,H)$ with $H = E$ as the subgraph with the desired properties.

For the induction step, suppose that for any digraph G with $|E| = m \ge \frac{2n\cdot(n-1)}{D+1}$ arcs exists a subgraph $G' = (V,H)$, $H \subseteq E$ with the desired properties.

Consider a graph $\bar{G} = (\bar{V},\bar{E})$ with $|\bar{E}| = m+1$ arcs. Since $m+1 > \frac{2n\cdot(n-1)}{D+1} \ge \bar{f}^{dir}(D,n)$, there exists an arc $e \in \bar{E}$ such that for any pair of vertices $u,w \in \bar{V}$ with $d_{\bar{G}}(u,w) \ge D$,

$$d_{\bar{G}_e}(u,w) = d_{\bar{G}}(u,w). \tag{11}$$

Note that the cardinality of the set of arcs of $\bar{G}_e$ is $|\bar{E} \setminus \{e\}| = |\bar{E}| - 1 = m$, and so the induction hypothesis is applicable to $\bar{G}_e$. In other words, there exists a subgraph $G' = (\bar{V},H)$ of $\bar{G}_e$, $H \subseteq \bar{E} \setminus \{e\} \subseteq \bar{E}$, with $|H| \le 2n(n-1)/(D+1)$, such that for any pair of vertices $u,w \in \bar{V}$ such that $d_{\bar{G}_e}(u,w) \ge D$,

$$d_{G'}(u,w) = d_{\bar{G}_e}(u,w). \tag{12}$$

Note that by (11), $d_{\bar{G}_e}(u,w) \ge D$ implies $d_{\bar{G}}(u,w) = d_{\bar{G}_e}(u,w) \ge D$, and so it follows that $G' = (\bar{V},H)$ is a subgraph of $\bar{G} = (\bar{V},\bar{E})$, $H \subseteq \bar{E}$, with $|H| \le 2n(n-1)/(D+1)$, such that for any pair of vertices $u,w \in V$ with $d_{\bar{G}}(u,w) = d_{\bar{G}_e}(u,w) \ge D$, $d_{G'}(u,w) = d_{\bar{G}_e}(u,w) = d_{\bar{G}}(u,w)$. The last two equalities are by (11) and (12).

Note that the edge e as above can be found in polynomial time, by computing all the distances in $\bar{G}_e$ for every $e \in \bar{E}$, and testing whether there is a pair of vertices $u,w \in \bar{V}$ such that $d_{\bar{G}}(u,w) \ge D$ and $d_{\bar{G}_e}(u,w) > d_{\bar{G}}(u,w)$.

Therefore, the entire computation of the subgraph G', that satisfies the assertion of the theorem, can be completed in polynomial time (specifically, in $O(|E|^3\cdot n)$ time).

Observe that if an edge e was examined once by the algorithm, and the algorithm decided not to remove it, it means that there is a pair of vertices u,w such that $d_{\bar{G}_e}(u,w) > d_{\bar{G}}(u,w) \ge D$. Consequently, for any subgraph D-preserver $\hat{G}$ of $\bar{G}$, $d_{\hat{G}_e}(u,w) \ge d_{\bar{G}_e}(u,w) > d_{\bar{G}}(u,w) \ge D$, and so, the edge e will never be removed by the algorithm. Hence the algorithm can examine each edge just once. This observation speeds the algorithm up by a factor of $|E|$, implying the desired running time of $O(|E|^2 n)$. □

We remark that after inequalities (7) and (8) were communicated to Mikkel Thorup, he devised [22] a more efficient randomized procedure for computing a subgraph D-preserver of size $O(n^2 \log n/D)$ (greater than optimal by a logarithmic factor). This more efficient procedure uses some techniques of [25] from the area of dynamic algorithms. The efficiency of the procedure of [22] makes it more suitable for algorithmic applications such as (and this is, indeed, the motivation of [22]) computing shortest paths between pairs of vertices that are at distance at least D one from another. We next use a similar idea to prove the existence of a distance labeling D-preserving scheme with labels of size $O((n/D)\cdot\log^2 n)$. This is tight up to a factor of $O(\log^2 n/\log D)$, in view of Corollary 2.16.

THEOREM 2.31. *For every integer number $D \ge 1$, there exists a distance labeling D-preserving scheme $(\mathcal{M},\mathcal{D})$ for a family of all (possibly directed) n-vertex unweighted graphs with labels of size $O((n/D)\cdot\log^2 n)$.*

Proof. Fix $2 < c < 3$ to be some real constant. Consider a labeling procedure that given an n-vertex graph $G = (V,E)$ starts with choosing a random subset $R \subseteq V$ of vertices. Every $v \in V$ is chosen into R independently at random with a probability of $p = \min\{c\log n/D, 1\}$.

Next, the procedure fixes an arbitrary ordering $(u_1, u_2, \ldots, u_{|R|})$ of the vertices of R. Then, for every pair of vertices $v \in V$, $u \in R$, the procedure forms a string $\alpha_v(u)$ to be the concatenation of the bit strings $d_G(v,u)$ and $d_G(u,v)$ (if the graph G is undirected, $\alpha(u)$ is the bit string representing $d_G(v,u) = d_G(u,v)$).

Finally, for every vertex $v \in V$, the procedure forms its label $\mathcal{M}(G,v)$ to be $\alpha_v(u_1) \cdot \alpha_v(u_2) \cdot \ldots \cdot \alpha_v(u_{|R|})$, where "$\cdot$" stands for concatenation.

Observe that $\mathbb{E}(|R|) = p \cdot n \leq c \log n \cdot n/D$. Hence, for every vertex $v \in V$, $|\mathcal{M}(G,v)| \leq c \log n \cdot n^2/D$. The query-answering procedure accepts as input two labels $\mathcal{M}(G,v_1) = \alpha_{v_1}(u_1) \cdot \alpha_{v_1}(u_2) \cdot \cdots \cdot \alpha_{v_1}(u_{|R|})$ and $\mathcal{M}(G,v_2) = \alpha_{v_2}(u_1) \cdot \alpha_{v_2}(u_2) \cdot \cdots \cdot \alpha_{v_2}(u_{|R|})$, and returns $\min\{d_G(v_1,u) + d_G(u,v_2) \mid u \in R\}$. Observe that for every $u \in R$, $d_G(v_i,u)$ can be computed given $\mathcal{M}(G,v_i)$, $i = 1,2$.

By Markov inequality,

$$\mathbb{P}(|R| \leq 2c \log n \cdot n/D) \geq 1/2. \tag{13}$$

For every pair of vertices (v_1,v_2), fix some shortest path P_{v_1,v_2} from v_1 to v_2 in G. (In an undirected graph, P_{v_1,v_2} coincides with P_{v_2,v_1}.) Observe that for v_1, v_2 such that $d_G(v_1,v_2) \geq D$, $|V(P_{v_1,v_2})| \geq D+1$. Note that for a vertex $z \in V(P_{v_1,v_2})$, $\mathbb{P}(z \in R) = c \log n/D$. Hence

$$\mathbb{P}(V(P_{v_1,v_2}) \cap R = \emptyset) \ = \ (1 - c\log n/D)^{D+1} \ \leq \ 1/n^c.$$

Hence,

$$\mathbb{P}(\exists v_1, v_2 \in V \ \text{ s.t. } \ d_G(v_1,v_2) \geq D \text{ and } V(P_{v_1,v_2}) \cap R = \emptyset) \ \leq \ n^2/n^c = 1/n^{c-2}.$$

In other words,

$$\mathbb{P}(\forall v_1, v_2 \in V \ \text{ s.t. } \ d_G(v_1,v_2) \geq D, V(P_{v_1,v_2}) \cap R \neq \emptyset) \ \geq \ 1 - 1/n^{c-2}.$$

Together with (13), this implies that

$$\mathbb{P}(|R| \leq 2c \log n \cdot n/D \text{ and } \forall v_1, v_2 \in V \ \text{ s.t. } \ d_G(v_1,v_2) \geq D, V(P_{v_1,v_2}) \cap R \neq \emptyset) \\ \geq 1/2 - 1/n^{c-2}.$$

Finally, note that the event $(\forall v \in V, |\mathcal{M}(G,v)| \leq 2c \log^2 n \cdot n/D)$ contains the event $(|R| \leq 2c \log n \cdot n/D)$, and for every pair of vertices $v_1, v_2 \in V$ the event $(V(P_{v_1,v_2}) \cap R \neq \emptyset)$ contains the event $(\mathcal{D}(\mathcal{M}(G,v_1), \mathcal{M}(G,v_2)) = d_G(v_1,v_2))$. Hence,

$$\mathbb{P}[\forall v \in V, |\mathcal{M}(G,v)| \leq 2c \log^2 n \cdot n/D, \text{ and } \forall v_1, v_2 \in V \ \text{ s.t. } \ d_G(v_1,v_2) \geq D, \\ \mathcal{D}(\mathcal{M}(G,v_1), \mathcal{M}(G,v_2)) = d_G(v_1,v_2)] \ \geq 1/2 - 1/n^{c-2} \ > \ 0,$$

for sufficiently large n.

Hence, there exists a D-preserving distance labeling scheme with labels of size $O(\log^2 n \cdot n/D)$. □

Next, we devise a polynomial time algorithm for constructing a diSteiner 1-preserver with $O(n^2/\log n)$ arcs for an arbitrary graph. In conjunction with Theorem 2.30, this yields a polynomial time algorithm for constructing a diSteiner D-preserver with $O(\frac{n^2 \log D}{D \log n})$ arcs for an arbitrary graph.

We remark that the main obstacle towards converting the proof of Corollary 2.29 into an efficient algorithm is the existential nature of the proof of Theorem 2.26. The next theorem is a constructive proof version of Theorem 2.26 and Lemma 2.25.

That is, an efficient algorithm for extracting a subgraph isomorphic to $K_{s,t}$ from a sufficiently dense graph. Another algorithm with a similar running time for extracting $K_{s,t}$ was devised by [16], and our algorithm is provided for completeness.

For any vertex $y \in V$, let $d(y)$ denote the degree of y.

THEOREM 2.32 (see [16]). *Let G be a graph of order n, $W \subseteq V(G)$, and $1 \leq s, t$. Suppose*

$$\sum_{y \in W} \binom{d(y)}{t} > (s-1)\binom{n}{t}. \tag{14}$$

Then G contains a $K_{s,t}$ with the "s part" contained in W, i.e., there are (necessarily disjoint) sets $S \subset W$ and $T \subset V$, $|S| = s$, $|T| = t$, such that every vertex of S is joined to every vertex of T. The $K_{s,t}$ can be computed in $O(n^2 \cdot t)$ time.

Proof. We shall do considerably more than claimed by the theorem. We shall give an algorithm that finds a "large" set $S \subset W$ all whose vertices are joined to all vertices of a set T with t vertices. Our condition (14) will imply that the set S constructed by the algorithm will have at least s vertices.

In our description of the algorithm, we shall say that a triple (G, W, t), with $W \subset V(G)$, is *s-large* if condition (14) is satisfied.

Here then is our plan. Starting with the triple (G, W, t), we perform the *t-step* of the algorithm to construct a vertex x_1 and a triple $(G_1, W_1, t-1)$, where $G_1 = G - x_1$, $W_1 \subset W \setminus \{x_1\}$, the vertex x_1 is joined to all vertices in W_1, and the triple $(G_1, W_1, t-1)$ is s-large, then perform the $(t-1)$-step of the algorithm to obtain a vertex $x_2 \in V(G_1)$ and a triple $(G_2, W_2, t-2)$ with $G_2 = G_1 - x_2$ and $W_2 \subset W_1 \setminus \{x_2\}$, such that x_2 is joined to every vertex in W_2, and the triple $(G_2, W_2, t-2)$ is s-large, and so on. Finally, after the 1-step of the algorithm, we get a vertex x_t and a triple $(G_t, W_t, 0)$. This completes the algorithm: our sets are $S = W_t$ and $T = \{x_1, x_2, \ldots, x_t\}$. By construction, G contains all edges from S to T and, as $(G_t, W_t, 0)$ is s-large, from (14) $\sum_{y \in W} \binom{d(y)}{t} > (s-1)\binom{n}{t}$ we shall find that $|S| \geq s$.

To complete our proof, here then is the t-step of the algorithm. For $x \in V(G)$, let the (t, W)-weight of x be

$$w(x) = w_{t,W}(x) = \sum_{(x,y) \in E,\ y \in W} \binom{d(y)-1}{t-1}.$$

Since

$$\sum_{x \in V} w(x) = \sum_{y \in W} d(y)\binom{d(y)-1}{t-1} = \sum_{y \in W} t\binom{d(y)}{t} > t(s-1)\binom{n}{t} = (s-1)n\binom{n-1}{t-1},$$

there is a vertex $x_1 \in V$ such that

$$\sum_{y \in W_1} \binom{d(y)-1}{t-1} > (s-1)\binom{n-1}{t-1}, \tag{15}$$

where $W_1 = \{y \in W : (x, y) \in E(G)\}$. Indeed, any vertex whose (t, W)-weight is at least the average will do for x_1; a vertex of maximal (t, W)-weight will certainly do. Set $G_1 = G - x_1$. Condition (15) means precisely that the triple $(G_1, W_1, t-1)$ is s-large (as for any $y \in W_1$, its degree in G_1 is $d(y) - 1$). Hence we can apply the

$(t-1)$-step of our algorithm to the triple $(G_1, W_1, t-1)$, and so on, until we get to an s-large triple $(G_t, W_t, 0)$. Since

$$|W_t| = \sum_{y \in W_t} \binom{d(y)-1}{0} > (s-1)\binom{n-1}{0} = s-1,$$

we find that $|W_t| \geq s$. By construction, the graph G contains all edges from $S = W_t$ to $T = \{x_1, x_2, \dots, x_t\}$.

A straightforward implementation of this algorithm requires $O(n^2 \cdot t)$ operations. Indeed, there are t iterations. On each iteration the algorithm chooses a vertex of minimal weight. It takes $O(|E|)$ operations to recompute the degrees, and $O(n)$ operations per vertex to compute its weight, summing up to an overall $O(n^2 + |E|) = O(n^2)$ operations per iteration. □

Corollary 2.33. *Let G be a graph of order n and size $nd/2$, i.e., average degree d. If $1 \leq t \leq s$ and*

$$n\binom{d}{t} > (s-1)\binom{n}{t}, \tag{16}$$

then G contains a $K_{s,t}$ subgraph. Furthermore, the algorithm described in the proof of Theorem 2.32 (starting with $W = V$) finds a $K_{s,t}$ subgraph.

Proof. Let G have degree sequence $(d_i)_1^n$. Then by the convexity of the binomial coefficient,

$$\sum_{i=1}^{n} \binom{d_i}{t} \geq n\binom{d}{t} > (s-1)\binom{n}{t}.$$

Hence, the result follows from Theorem 2.32. □

Remark. In applying Corollary 2.33, we should always assume that $s \geq t$ since if (16) holds for $s \leq t$, then it also holds when s and t are interchanged.

Observe that it follows that $K_{s,2}$ can be extracted from an n-vertex m-edge graph G whenever $m \geq c \cdot \sqrt{s} n^{3/2}$ for some universal constant c. Under this condition, whenever $s \geq 3$, it is possible to construct a diSteiner 1-preserver for G with $m' < m$ edges.

Corollary 2.34. *Let G be a bipartite graph with bipartition (W, U), where $|U| = n$. If*

$$\sum_{y \in W} \binom{d(y)}{t} > (s-1)\binom{n}{t},$$

then G contains a $K_{s,t}$ subgraph, with s vertices in W and t in U.

The next corollary is a constructive analogue of Lemma 2.25.

Corollary 2.35. *There is an algorithm that given an n-vertex graph $G = (V, E)$ computes a subgraph of G isomorphic to $K_{a,a}$ with $a = \Omega(\frac{\log n}{\log(n^2/|E|)})$ in $O(n^2 \cdot \frac{\log n}{\log(n^2/|E|)})$ time.*

The next theorem addresses the question of constructibility of sparse diSteiner 1-preservers for arbitrary graphs.

Theorem 2.36. *For every n-vertex (di)graph, a diSteiner 1-preserver with $O(n^2/\log n)$ arcs of weight 1 or 1/2 can be constructed in $O(n^4 \cdot \frac{(\log\log n)^2}{\log n})$ time.*

Proof. To construct a diSteiner 1-preserver with at most $O(n^2/\log n)$ arcs for an arbitrary (di)graph, one needs to invoke the procedure of extracting $K_{a,a}$ at most $O(n^2 \log\log n/\log^2 n)$ times. Indeed, in a graph with $m = \Omega(n^2/\log n)$ edges, $a = \Omega(\log n/\log(n^2/m)) = \Omega(\log n/\log\log n)$, and so a single extraction of $K_{a,a}$ results in eliminating $\Omega(\log^2 n/(\log\log n)^2)$ edges from the graph. As we start with $O(n^2)$ edges, after $O(\frac{n^2(\log\log n)^2}{\log^2 n})$ extractions, the number of edges left in the graph is $O(n^2/\log n)$. By Corollary 2.35, each extraction can be completed in $O(n^2 \cdot \log n)$ time, and so, the assertion of the theorem follows. □

We remark that any improvement of a factor of $\Omega(n)$ of the running time in Theorem 2.36 to $o(|E|\cdot n)$ would have some interesting applications to efficient computation of distances in dense graphs (by computing their diSteiner 1-preserver, and performing distance computations on the 1-preserver, assuming that the latter is sparser than the original graph).

Next, observe that a polynomial time algorithm for constructing diSteiner D-preserver for an arbitrary (di)graph can be obtained by composing the results of Theorems 2.30 and 2.36.

COROLLARY 2.37. *For any n-vertex (di)graph $G = (V,E)$ and an integer $D \geq 1$, a diSteiner D-preserver with $O(n^2/\log n)$ arcs of weight 1 or $1/2$ can be constructed in $O(|E|^2 \cdot n + n^4 \cdot \frac{(\log\log n)^2}{\log n})$ time.*

2.3.3. (D, g)-preservers.

Next, we present an upper bound on $\bar{f}(D,g,n)$, that is, the size of the n-vertex extremal graph whose only (D,g)-preserver is the graph itself.

Recall that our upper bound on $f(D,n)$, that is, the minimal value such that any n-vertex graph has a D-preserver with at most $f(D,n)$ edges, was derived through the analysis of the size of the extremal graph G whose only subgraph D-preserver is G itself, i.e., $\bar{f}(D,n)$. This was possible due to the duality $f(D,n) = \bar{f}(D,n)$ (Lemma 2.4). In the case of (D,g)-preservers we are not aware of any upper bound on $f(D,g,n)$ in terms of $\bar{f}(D,g,n)$. However, we believe that the bounds on $\bar{f}(D,g,n)$ are of independent interest, and may also serve as a first step towards a better understanding the behavior of $f(D,g,n)$.

DEFINITION 2.38. *In an undirected graph $G = (V,E)$, a sequence of distinct vertices $C = (v_0, v_1, \ldots, v_s, v_0)$ is a* cycle, *if $v_i \in V$ for any index i $0 \leq i \leq s$, $v_i \in V$, and for any index i, $0 \leq i \leq s-1$, $(v_i, v_{i+1}) \in E$ and $(v_s, v_0) \in E$. The* length *of the cycle C is $s+1$. The* girth *of a graph G is the length of the shortest cycle of G.*

The following observation is derived directly from the definition of (D,g)-preserver.

LEMMA 2.39. *Every graph $G = (V,E)$ whose only (D,g)-preserver is G itself satisfies $girth(G) \geq g+2$.*

Proof. Suppose for contradiction that $girth(G) \leq g+1$.

Then there exists an edge $e = (u,w)$ such that $d_{G_e}(u,w) \leq g$. Since G_e is not a (D,g)-preserver of G there exists a pair of vertices $x,y \in V$ such that $d_G(x,y) \geq D$, and

$$d_{G_e}(x,y) \geq d_G(x,y) + g. \tag{17}$$

Let P be one of the shortest paths from x to y in G. Obviously, the edge e belongs to P. In other words, without loss of generality, $P = (x = v_0, \ldots, v_t = u, v_{t+1} = w, \ldots, v_s = y)$, for $|P| = s$, $t = 0,1,\ldots,s-1$. Let P_1 be one of the shortest paths from u to w in G_e. Note that $|P_1| = d_{G_e}(u,w) \leq g$. Let $P_{x,u}$ denote the path $(x = v_0, v_1, \ldots, v_t = u)$, and $P_{w,y}$ denote the path $(v_{t+1} = w, v_{t+2}, \ldots, v_s = y)$.

Consider the walk $P_2 = P_{x,u} \cdot P_1 \cdot P_{w,y}$. Also, $|P_2| = |P_{x,u}| + |P_1| + |P_{w,y}| \leq t + g + s - (t+1) = s + g - 1 = |P| + g - 1 = d_G(x,y) + g - 1$. Note that $P_2 \subseteq E \setminus \{e\}$ is a path between x and y. Thus, $d_G(x,y) + g - 1 \geq |P_2| \geq d_{G_e}(x,y)$. However, this contradicts (17). $\square$

Recall that for any integer $r \geq 3$, any n-vertex graph $G = (V,E)$ of girth at least r has at most $(n^{1+1/r-2} + n)$ edges (cf. [8]). Therefore, Lemma 2.39 implies that $\bar{f}(D,g,n) \leq n^{1+2/g} + n$. We next establish another upper bound on $\bar{f}(D,g,n)$, which is tighter whenever $D = \Omega(\sqrt{n})$.

Definition 2.40. *For a graph $G = (V,E)$, a vertex $v \in V$, and integer $k = 0,1,2,\ldots$, let $\Gamma_k(v,G)$ (resp., $\hat{\Gamma}_k(v,G)$) denote the set of vertices that are at distance precisely (resp., at most) k from v, i.e., $\Gamma_k(v,G) = \{u \in V \mid d_G(v,u) = k\}$, $\hat{\Gamma}_k(v,G) = \{u \in V \mid d_G(v,u) \leq k\}$.*

Theorem 2.41. *For integer numbers D, g, and n, $D \geq 2$, $g \geq 8$, and n sufficiently large, $\bar{f}(D,g,n) \leq 4n^{1+1/\lfloor g/4 \rfloor}/D^{1/\lfloor g/4 \rfloor}$.*

Proof. For every edge $e = (u,w) \in E$, let $P(e)$ be one of the shortest paths between $endpoint(P(e),e,u)$ and $endpoint(P(e),e,w)$ in G such that $d_G(endpoint(P(e),e,u),\ endpoint(P(e),e,w)) \geq D$, but

$$\begin{aligned} &d_{G_e}(endpoint(P(e),e,u), endpoint(P(e),e,w)) \ > \\ &d_G(endpoint(P(e),e,u), endpoint(P(e),e,w)) + g. \end{aligned}$$

Note that $|subpath(P(e),e,u)| + |subpath(P(e),e,w)| \geq D - 1$.

Let $long\_subpath(P(e), e = (u,w))$ denote the longer path among $subpath(P(e),e,u)$ and $subpath(P(e),e,w)$ (if they are equal choose one of them arbitrarily).

Note that for any edge $e \in E$,

$$(18) \qquad |long\_subpath(P(e),e)| \geq \lceil (D-1)/2 \rceil \geq D/2 - 1.$$

For a vertex $v \in V$, let

$$S(v) \ = \ \{e = (v,z) \in E \mid long\_subpath(P(e),e) = subpath(P(e),e,z)\}.$$

Consider some vertex $u \in \hat{\Gamma}_{\lfloor g/4 \rfloor - 1}(v,G)$. Let $S(u,v) = \{e = (u,z) \in E \mid d_G(v,z) = d_G(v,u) + 1, long\_subpath(P(e),e) = subpath(P(e),e,z)\}$.

Note that $S(v) = S(v,v)$. Note also that since $girth(G) \geq g+2$, and $d_G(v,u) \leq \lfloor g/4 \rfloor - 1$, it follows that for any edge $(u,z) \in S(u,v)$, the only shortest path from v to z in G passes through u.

Let $\hat{S}(v)$ denote the set $\bigcup_{u \in \hat{\Gamma}_{(\lfloor g/4 \rfloor - 1)}(v,G)} S(u,v)$. Let $\hat{P}(v)$ denote the set

$$(19) \qquad \hat{P}(v) \ = \ \{long\_subpath(P(e),e) \mid e \in \hat{S}(v)\}.$$

Next, we argue that for any two paths $P_1, P_2 \in \hat{P}(v)$, $V(P_1) \cap V(P_2) = \emptyset$. Let $e_1 = (u_1,z_1)$ be an edge of P_1, and $e_1 = (u_2,z_2)$ be an edge of P_2. Denote $x_1 = endpoint(P_1,(u_1,z_1),u_1)$, $x_2 = endpoint(P_2,(u_2,z_2),u_2)$, $y_1 = endpoint(P_1,(u_1,z_1),z_1)$, $y_2 = endpoint(P_2,(u_2,z_2),z_2)$.

Suppose for contradiction that there exists a vertex w such that $w \in V(P_1) \cap V(P_2)$.

Denote the segments of P_1 (resp., P_2) from x_1 (resp., x_2) to u_1 (resp., u_2), from u_1 (resp., u_2) to w, and from w to y_1 (resp., y_2), by P_1', P_1'' and P_1''' (resp., P_2', P_2'', and P_2'''), respectively.

Next, we show that

$$d_G(u_2, w) - (g/2 - 2) \leq d_G(u_1, w) \leq d_G(u_2, w) + (g/2 - 2). \tag{20}$$

Indeed, suppose for contradiction that $d_G(u_1, w) < d_G(u_2, w) - (g/2 - 2)$ (the case of $d_G(u_1, w) > d_G(u_2, w) + (g/2 - 2)$ is symmetrical).

Thus,

$$d_G(u_1, w) + (g/2 - 2) < d_G(u_2, w). \tag{21}$$

Then consider the path $P_{u_2,w} = P_{u_2,v} \cdot P_{v,u_1} \cdot P_1''$, where $P_{u_2,v}$ is the shortest path from u_2 to v in G, and P_{v,u_1} is the shortest path from v to u_1 in G.

Note that

$$\begin{aligned} |P_{u_2,w}| &= |P_{u_2,v}| + |P_{v,u_1}| + |P_1''| \\ &\leq 2(g/4 - 1) + d_G(u_1, w) \;=\; d_G(u_1, w) + g/2 - 2 \;<\; d_G(u_2, w) \end{aligned}$$

(the last inequality is by (21)). This is a contradiction, since $P_{u_2,w}$ is a path from u_2 to w. Hence, (20) follows.

Note that $P_{u_2,v}, P_{v,u_1} \subseteq E \backslash \{e_1\}$. Consider the path $P_{12} = P_1' \cdot P_{u_1,v} \cdot P_{v,u_2} \cdot P_2'' \cdot P_1'''$. Note that P_{12} is a path between x_1 and y_1 in G_{e_1}, since G satisfies the large-error property,

$$\begin{aligned} |P_{12}| &= |P_1'| + |P_{u_1,v}| + |P_{v,u_2}| + |P_2''| + |P_1'''| \\ &\geq d_{G_{e_1}}(x_1, y_1) \;\geq\; d_G(x_1, y_1) + g \;=\; |P_1'| + |P_1''| + |P_1'''| + g. \end{aligned}$$

That is, $|P_{u_1,v}| + |P_{v,u_2}| + |P_2''| \geq |P_1''| + g$.

Recall that $|P_{u_1,v}| + |P_{v,u_2}| \leq g/2 - 2$. Thus,

$$|P_2''| + (g/2 - 2) \geq |P_{u_1,v}| + |P_{v,u_2}| + |P_2''| \geq |P_1''| + g.$$

That is, $|P_2''| \geq |P_1''| + (g/2 + 2)$. In other words, $d_G(u_2, w) \geq d_G(u_1, w) + (g/2 + 2)$, contradicting (20).

Thus, $V(P_1) \cap V(P_2) = \emptyset$. Hence, the set $\hat{P}(v)$, defined by (19), consists of vertex-disjoint paths.

Thus, for any vertex $v \in V$,

$$\sum_{e \in \hat{S}(v)} |V(\mathit{long\_subpath}(P(e), e))| \leq n.$$

Hence,

$$\sum_{v \in V} \sum_{e \in \hat{S}(v)} |V(\mathit{long\_subpath}(P(e), e))| \leq n^2.$$

Using (18) it follows that

$$\sum_{v \in V} |\hat{S}(v)| \leq 2n^2/D. \tag{22}$$

Consider a digraph $\hat{G} = (V, \hat{E})$ with the same vertex set V as the graph G, but

$$\begin{aligned} \hat{E} \;=\; \{\langle u, w\rangle \mid\ & (u, w) \in E, \\ & \mathit{long\_subpath}(P((u, w)), (u, w)) = \mathit{subpath}(P((u, w)), (u, w), w)\}. \end{aligned}$$

In other words, every edge e of the graph G is oriented towards the endpoint w from which the subpath $\mathit{subpath}(P(e), e, w)$ is longer.

Observe that

$$\hat{S}(v) = \{e = (u, z) \mid \langle u, z\rangle \in \hat{E}, d_{G_e}(v, u) \le \lfloor g/4\rfloor - 1\}.$$

Let $a_0 = 2|E|/n$ be the average degree in G. Set $C = \lfloor a_0/4\rfloor = \lfloor |E|/2n\rfloor$. We construct a graph $G' = (V', E')$ in the following way. While there is a vertex $v \in V$ with $\mathit{deg}_G(v) \le C$, remove v from V and all its incident edges.

Note that at most $C \cdot n \le |E|/2$ edges are removed. That is, $|E'| \ge |E|/2$. Also, for any vertex $v \in V'$, $\mathit{deg}_{G'}(v) \ge C+1 \ge |E|/2n$. Also, $\mathit{girth}(G') \ge \mathit{girth}(G) \ge g+2$.

Consider,

$$\hat{S}'(v) \;=\; \{e = (u, z) \in E' \mid \langle u, z\rangle \in \hat{E}, d_{G'_e}(v, u) \le \lfloor g/4\rfloor - 1\}.$$

Note that for any vertex $v \in V'$, $\hat{S}'(v) \subseteq \hat{S}(v)$. Hence $\sum_{v\in V'} |\hat{S}'(v)| \le 2n^2/D$.

Note that for any edge $e = (u, z) \in E'$ either $\langle u, z\rangle \in \hat{E}$ or $\langle z, u\rangle \in \hat{E}$. For any edge $e = (u, z) \in E'$, denote

$$\mathit{far\_endpoint}(e) = \begin{cases} u, & \langle u, z\rangle \in \hat{E},\\ z, & \langle z, u\rangle \in \hat{E}.\end{cases}$$

Note that

$$\sum_{v\in V'} |\hat{S}'(v)| \;=\; \sum_{e\in E'} |\hat{\Gamma}_{\lfloor g/4\rfloor -1}(\mathit{far\_endpoint}(e), G'_e)|.$$

Note that since the minimal degree in G'_e is at least $|E|/2n$, and $\mathit{girth}(G'_e) \ge g+2$, it follows that for any edge $e \in E'$,

$$|\hat{\Gamma}_{\lfloor g/4\rfloor -1}(\mathit{far\_endpoint}(e), G'_e)| \;\ge\; (|E|/2n - 1)^{\lfloor g/4\rfloor -1}.$$

Therefore,

$$\begin{aligned}\sum_{v\in V'} |\hat{S}'(v)| &= \sum_{e\in E'} |\hat{\Gamma}_{\lfloor g/4\rfloor -1}(\mathit{far\_endpoint}(e), G'_e)|\\ &\ge |E'| \cdot (|E|/2n - 1)^{\lfloor g/4\rfloor -1}.\end{aligned}$$

By Theorem 2.20, we can assume that $|E| \ge 4n$. Hence,

$$\sum_{v\in V'} |\hat{S}'(v)| \;\ge\; |E|/2 \cdot \frac{|E|^{\lfloor g/4\rfloor -1}}{n^{\lfloor g/4\rfloor -1}\cdot 4^{\lfloor g/4\rfloor -1}} \;\ge\; \frac{|E|^{\lfloor g/4\rfloor}}{n^{\lfloor g/4\rfloor -1}\cdot 2\cdot 4^{\lfloor g/4\rfloor -1}}.$$

Hence, by (22),

$$\frac{2n^2}{D} \ge \frac{|E|^{\lfloor g/4\rfloor}}{n^{\lfloor g/4\rfloor -1}\cdot 2\cdot 4^{\lfloor g/4\rfloor -1}}.$$

Hence, for a sufficiently large n, it follows that $|E|^{\lfloor g/4\rfloor} D \le 4^{\lfloor g/4\rfloor} \cdot n^{\lfloor g/4\rfloor +1}$. Thus, $|E| \cdot D^{1/\lfloor g/4\rfloor} \;\le\; 4 \cdot n^{1+1/\lfloor g/4\rfloor}$. □

Acknowledgments. The third author is grateful to David Peleg, Mikkel Thorup, Mike Saks, Uri Zwick, Cyril Gavoile, Michael Langberg, and Tadao Takaoka for helpful discussions, and to an anonymous referee for pointing out reference [16]. The authors are also grateful to an anonymous referee for multiple remarks that helped to drastically improve the presentation of this paper, and for pointing out that the running time in Theorem 2.30 can be improved from $O(|E|^3 n)$ to $O(|E|^2 n)$.

REFERENCES

[1] I. Althöfer, G. Das, D. Dobkin, D. Joseph, and J. Soares, *On sparse spanners of weighted graphs*, Discrete Comput. Geom., 9 (1993), pp. 81–100.

[2] B. Awerbuch and D. Peleg, *Sparse partitions*, in Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science, 1990, pp. 503–513.

[3] B. Awerbuch and D. Peleg, *Network synchronization with polylogarithmic overhead*, in Proceedings of the 31st Annual Symposium on Foundations of Computer Science, IEEE, 1990, pp. 514–522.

[4] B. Awerbuch, B. Berger, L. Cowen, and D. Peleg, *Near-linear time construction of sparse neighborhood covers*, SIAM J. Comput., 28 (1998), pp. 263–277.

[5] B. Bollobás, *Modern Graph Theory*, Graduate Texts in Mathematics 184, Springer-Verlag, New York, 1998, pp. xiv+394.

[6] B. Bollobás, D. Coppersmith, and M. L. Elkin, *Sparse distance preservers and additive spanners*, in Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, Baltimore, MD, 2003, pp. 414–423.

[7] S. Baswana, T. Kavitha, K. Mehlhorn, and S. Pettie, *New constructions of (α, β)-spanners and purely additive spanners*, in Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, Vancouver, BC, Canada, 2005, pp. 672–681.

[8] J. A. Bondy and M. Simonovits, *Cycles of even length in graphs*, J. Combin. Theory Ser. B, 16 (1974), pp. 97–105.

[9] B. Chandra, G. Das, G. Narasimhan, and J. Soares, *New sparseness results on graph spanners*, in Proceedings of the 8th Annual ACM Symposium on Computational Geometry, Berlin, 1992, pp. 192–201.

[10] L. P. Chew, *There is a planar graph almost as good as the complete graph*, in 2nd Annual Symposium on Computational Geometry, Yorktown Heights, NY, 1986, pp. 169–177.

[11] E. Cohen, *Fast Algorithms for constructing t-spanners and paths of stretch t*, in Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science, Palo Alto, CA, pp. 648–658.

[12] E. Cohen, *Polylog-time and near-linear work approximation scheme for undirected shortest paths*, in Proceedings of the 26th Annual ACM Symposium on Theory of Computation, Montreal, Canada, 1994, pp. 16–26.

[13] D. P. Dobkin, S. J. Friedman, and K. J. Supowit, *Delaunay graphs are almost as good as complete graphs*, Discrete Comput. Geom., 5 (1990), pp. 399–407.

[14] M. L. Elkin, *Computing almost shortest paths*, in Proceedings of the 20th Annual ACM Symposium on Principles of Distributed Computing, Newport, RI, 2001, pp. 53–63.

[15] M. L. Elkin and D. Peleg, *$(1+\epsilon, \beta)$-spanner constructions for general graphs*, SIAM J. Comput., 33 (2004), pp. 608–631. See also Proc. 33rd ACM Symp. on Theory of Computing, Crete, Greece, 2001, pp. 173–182.

[16] T. Feder and R. Motwani, *Clique partitions, graph compression and speeding-up algorithms*, Special Issue for the STOC conference, J. Comput. System Sci., 51 (1995), pp. 261–272.

[17] C. Gavoille, D. Peleg, S. Perennes, and R. Raz, *Distance labeling in graphs*, in Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms, Washington, DC, 2001, pp. 210–219.

[18] R. L. Graham, B. L. Rothschield, and J. H. Spencer, *Ramsey Theory*, 2nd ed., Wiley-Interscience Series in Discrete Mathematics and Optimization, A Wiley-Interscience Publication, John Wiley & Sons, Inc., New York, 1990, pp. xii+196.

[19] D. Peleg, *Proximity-preserving labeling schemes*, J. Graph Theory, 33 (2000), pp. 167–176.

[20] D. Peleg and A. Schäffer, *Graph spanners*, J. Graph Theory, 13 (1989), pp. 99–116.

[21] D. Peleg and J. D. Ullman, *An optimal synchronizer for the hypercube*, SIAM J. Comput., 18 (1989), pp. 740–747.

[22] M. Thorup, *private communication*, Oct. 2001.

[23] M. Thorup, *Compact oracles for reachability and approximate distances in planar digraphs*,

in Proceedings of the 42nd Annual Symposium on Foundations of Computer Science, Las Vegas, NV, 2001.

[24] M. Thorup and U. Zwick, *Approximate distance oracles*, in Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, Crete, Greece, 2001, pp. 183–192.

[25] J. D. Ullman and M. Yannakakis, *High-probability parallel transitive-closure-algorithms*, SIAM J. Comput., 20 (1991), pp. 100–125.

SIAM J. Discrete Math.
Vol. 19, No. 4, pp. 1056–1064

THE TWO-BATCH LIAR GAME OVER AN ARBITRARY CHANNEL*

IOANA DUMITRIU† AND JOEL SPENCER‡

Abstract. We consider liar games in which player Paul must ask one full batch of questions, receive all answers, and then ask a second and final batch of questions. We show that the effect of this restriction is asymptotically negligible. The strategy for Paul is given explicitly.

Key words. liar games, coding theory, channel, asymptotics, adaptive strategy

AMS subject classifications. 91A05, 91A46, 94B25,68R05, 68W01

DOI. 10.1137/040617510

1. Introduction.

In this paper, we present a variant of the Rényi–Ulam game which is similar to the one we considered in [5]. The main difference consists in the type of strategy Paul is allowed to employ; the case we study here has Paul using a semi-offline strategy, as opposed to the completely online one of [5]. We introduce the game and present a few variants and recent results, comparing and contrasting them to the results in our current work.

1.1. History and recent results.

In the original two-player Rényi–Ulam game, Carole (player 1) thinks of a number $x \in \{1, \dots, n\}$, while Paul (player 2) must find it by asking q Yes/No questions. The catch is that Carole is allowed to lie, but only at most k times (k being a fixed integer). The question is "for which n, q, k can Paul guess the number and win?"

Many researchers have examined this game and variants thereof; there is an extensive literature on the subject, of which we mention Pelc's excellent survey article [6]. For the reader interested in the history of the subject, good references are provided in Rényi [7], Ulam [10], and Berlekamp [2].

Historically, the full-lie version (where Carole is allowed to lie in whichever way she chooses, when she chooses to do it) was considered first; it is known that for this case, Carole can win when

$$2^q < n \left(\sum_{i=0}^{k} \binom{q}{i} \right),$$

and the converse is roughly true when n and q are large (while k is fixed; see [8]).

The more recently introduced half-lie case restricts Carole's ability to lie by requiring her to tell the truth when the truthful answer is Yes. Cicalese and Mundici [3] have shown that in the one half-lie case ($k = 1$), the maximal n for which Paul can win with q questions is asymptotic to $\frac{2^{q+1}}{q}$, as q goes to infinity.

In our earlier paper [4], we have shown that for any fixed number k of half-lies, the asymptotics for the maximal value of n as a function of q and k as q goes to infinity are given by $\frac{2^{q+k}}{\binom{q}{k}}$.

*Received by the editors October 22, 2004; accepted for publication (in revised form) September 26, 2005; published electronically February 3, 2006.

http://www.siam.org/journals/sidma/19-4/61751.html

†Miller Institute and U.C. Berkeley, Department of Mathematics, Berkeley, CA 94720 (dumitriu@math.berkeley.edu). This research was completed while this author was a Miller Research Fellow in the Department of Mathematics at U.C. Berkeley.

‡Courant Institute of Mathematical Sciences, New York, NY 10012 (spencer@cims.nyu.edu).

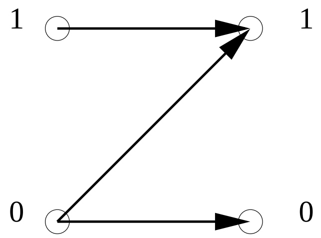


FIG. 1. *The Z-channel.*

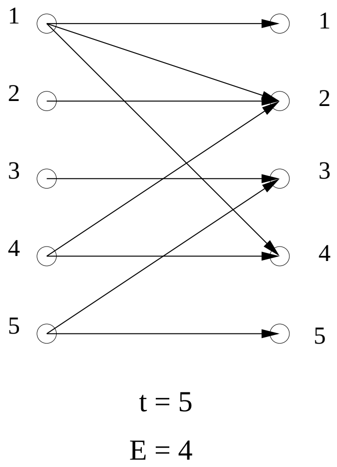


FIG. 2. *A* 5*-ary channel* C*, with* $E = 4$*.*

There is a simple connection between the half-lie case and the Z-channel of coding theory, where during communication, a 0 can be accidentally transformed into a 1, but a 1 has to be always transmitted as a 1. (We allow for false positives, but not for false negatives; see Figure 1.)

Following this idea, we have further extended our results in [5], where we found asymptotics for arbitrary channels C.

Below we give a few definitions (which will be referred to throughout this paper) and state the main theorem from [5].

DEFINITION 1. *A* t*-ary channel* C *is a set of ordered pairs* (x, y) *with* $1 \leq x, y \leq t$*, both integers, such that for each* $1 \leq x \leq t$*,* $(x, x) \in C$*. The pairs* $(x, y) \in C$ *with* $y \neq x$ *are called potential errors. The total number of potential errors is denoted by* E*.*

Figure 2 is an "arbitrary" channel, which we also used in [5].

We now define (from [5]) the focus of our work, the (n, k, C)-liar game with q questions. There are two players, Paul and Carole, and q rounds. There is a set Ω of size n, the possibilities. Carole thinks of an $\alpha \in \Omega$. On each round Paul partitions Ω into disjoint sets $A_1, \ldots, A_t$; Carole finds that i for which $\alpha \in A_i$ and responds with either i or some $j \neq i$ with $(i, j) \in C$. The latter case is called a lie or an error; we use the terms interchangably in this work. (We note that in the coding theory literature one wishes to send information through a channel and one wishes that it be deciphered accurately even when there is a certain number E of errors. These are called E-error correcting codes.)

Paul's choice of partitions in later rounds can, and in general will, depend on Carole's responses—hence we say that the channel allows for feedback.

Carole can make at most k lies in the course of the game. At the end of the q rounds Paul has won if and only if there is only one possible $\alpha \in \Omega$ for which Carole could have made her responses.

DEFINITION 2. *Let $A_{C,k}(q)$ be the maximal n such that there is a winning strategy for Paul in the (n, k, C) game with q questions.*

THEOREM 1.1 (main result from [5]). *Let C be an arbitrary (fixed) t-ary channel with $E > 0$ potential errors. Then for any fixed $k \in \mathbb{N}$,*

$$A_{C,k}(q) \sim \left(\frac{t}{E}\right)^k \frac{t^q}{\binom{q}{k}},$$

where the asymptotics are taken as $q \to \infty$.

We recall the vector format of [5].

DEFINITION 3. *Consider an intermediate position in the game. For $0 \le i \le k$ let x_i be the number of possibilities for which Carole has lied precisely i times. We call the vector $\mathbf{x} = (x_0, \ldots, x_k)$ the* state vector.

We remark that the state vector completely characterizes the state of the game, up to a renaming of Carole's possible choices. We further allow a game to have starting state $\mathbf{x} = (x_0, \ldots, x_k)$. In such a game Carole has x_j possibilities for which she may lie $(k-j)$ times, $0 \le j \le k$.

1.2. Two batch strategies and the main result.

In the coding theory setting described in the previous section, we have the following problem: Bob sends $x \in \{1, \ldots, t\}^q$ to Alice through channel C, and the channel may make as many as k mistakes. Bob's full message is one of n possibilities. Is there a protocol by which correct reception of the message by Alice is ensured?

The answer is yes if and only if Paul can win the (n, k, C) game with q questions *by asking all the questions in advance*. This is an additional and very strong constraint imposed on the game we have described in the previous section, which assumes complete feedback (each question is followed by an answer and each question is based on all previous answers, in a completely online strategy). The offline, no-feedback constraint appears to change the problem drastically. We denote by $A^-_{C,k}(q)$ the maximal n for which Paul wins under the offline constraint. Clearly $A^-_{C,k}(q) \le A_{C,k}(q)$. However, the asymptotics of $A^-_{C,k}(q)$ (indeed, even those of $A^-_{C,1}(q)$) remain a challenging open question.

In this work we consider a *two-batch* strategy for Paul. In this variant of the game, Paul is constrained to first ask a batch of q_1 questions (all at once, offline). After listening to Carole's answers, he asks a second batch of $q_2 = q - q_1$ offline questions. Finally, depending on the answers to the second batch of questions, Paul wins if he is certain which one of the n possibilities Carole had in mind. We allow Paul to determine q_1.

This constraint that we impose on Paul naturally makes it harder for him to win. Denote by $\tilde{A}_{C,k}(q)$ the maximal n for which Paul wins the two-batch liar game over the channel C with q questions and up to k lies. Then $\tilde{A}_{C,k}(q) \le A_{C,k}(q)$, so an immediate upper bound for the asymptotics of $\tilde{A}_{C,k}(q)$ is given by Theorem 1.1:

$$\tilde{A}_{C,k}(q) \le (1 + o(1)) \left(\frac{t}{E}\right)^k \frac{t^q}{\binom{q}{k}}.$$

When we first considered the two-batch variant of the game, we expected that the asymptotics were going to be significantly smaller than in the online case. To our surprise, the asymptotics turned out to coincide. We summarize this in the statement of this paper's main result.

THEOREM 1.2 (main result). *Let C be an arbitrary (fixed) t-ary channels with $E > 0$ potential errors. Then for any fixed $k \in \mathbb{N}$,*

$$\tilde{A}_{C,k}(q) \sim \left(\frac{t}{E}\right)^k \frac{t^q}{\binom{q}{k}},$$

where the asymptotics are taken as $q \to \infty$.

1.3. Main idea. We sketch here the main idea of the proof of Theorem 1.2.

We will consider a strategy for Paul which will ask almost all questions in the first batch; asymptotically, we will have $q_1 \sim q$, while $q_2 \sim k \log q$. This means that Paul will ask most of the questions offline, at first, narrowing the possibilities in a way that will make it possible for him to use in the second batch a logarithmically smaller number of questions in order to separate the right answer.

As seen in the previous section, an asymptotic upper bound for $\tilde{A}_{C,k}(q)$ is represented by the asymptotics for $A_{C,k}(q)$, since restricting Paul to a two-batch strategy makes things harder for him. To prove Theorem 1.2, we thus need only to prove that $\left(\frac{t}{E}\right)^k \frac{t^q}{\binom{q}{k}}$ is an asymptotic lower bound for $\tilde{A}_{C,k}(q)$. In other words, the essential element of our current work is to give an effective strategy for Paul.

To do this we will prove that for any $\alpha < \left(\frac{t}{E}\right)^k$, there is a q_0 large enough so that for every $q \geq q_0$ and every $n < \alpha \frac{t^q}{\binom{q}{k}}$ Paul can win the two-batch (n, k, C) game.

As in [5], we will first basically reduce the problem to considering n of the form $(1-\delta)at^s$ with δ a small positive constant and a a positive integer of bounded size. The heart of the argument is the notion of *balanced strings* of length s, strings from the channel alphabet $\mathcal{A}$ (usually $\mathcal{A} = \{1, \dots, t\}$) in which each letter appears roughly the same number of times. The possible answers of Carole are then represented as pairs $(i, u_1 \cdots u_s)$ with $1 \leq i \leq a$ and $u_1 \cdots u_s$ one of these balanced strings. Paul then asks for the values of $u_1, \dots, u_s$. This strategy *forces* Carole to give each of the t possible replies roughly the same number of times. Carole is thus, roughly speaking, precluded from taking advantage of the asymmetries of the channel.

For the endgame, we will employ the same type of "packing is winning" argument that we have used in [5].

2. Preliminaries.

2.1. Balanced strings and $(1-\delta)$.

DEFINITION 4. *Given $s, t \in \mathbb{N}$ and $r > 0$, we call a string of length s letters from the alphabet $\mathcal{A} = \{1, 2, \dots, t\}$ r-balanced if for every $i \in \mathcal{A}$, the number of occurrences of the letter i is at most $s/t + r$.*

We let $\mathcal{B}_{s,t}^{r}$ denote the set of such r-balanced strings.

LEMMA 2.1. *Given $t \in \mathbb{N}, \delta > 0$, there exists a s_0 such that for all $s \geq s_0$, for $r = s^{2/3}$, the number of r-balanced strings of length s with letters from the alphabet $\mathcal{A}$ is at least $(1-\delta)t^s$.*

Proof. For any particular letter i, the number of strings where that letter appears more than $s/t + r$ times is t^s times $\Pr[B[s, 1/t] > s/t + r]$, where B denotes the binomial distribution. A standard second moment method (see, e.g., Theorem A.1.11

of [1] for tighter Chernoff bounds) gives that this probability is $o(1)$, where t is fixed and the asymptotics are as $s \to \infty$.

There is a fixed number t of letters, so the probability that any one of them appears more than $s/t + r$ times in a string of length s is still $o(1)$. Thus, the number of r-balanced strings is at least $(1 - o(1))t^s$.

By taking s sufficiently large this is at least $(1 - \delta)t^s$. □

2.2. Choosing the density parameter *a*. We show in this section that it is enough to consider numbers of the type $\lfloor (1-\delta)at^s \rfloor$, where $a \in (t^T, t^{T+1}] \cap \mathbb{N}$ with T an integer that depends only on δ and t.

LEMMA 2.2. *Given t, E, k, $\delta \in (0,1)$ and given any $0 < \alpha < \alpha' < (t/E)^k$, there exists $T \in \mathbb{N}$ and $q_0 \in \mathbb{N}$ such that for any $q \geq q_0$, for any $n \leq \alpha \frac{t^q}{\binom{q}{k}}$, there exist $a \in (t^T, t^{T+1}] \cap \mathbb{N}$ and nonnegative integer s such that*

$$n \leq (1-\delta)at^s < \alpha' \frac{t^q}{\binom{q}{k}}.$$

Proof. With foreknowledge, we let T be such that

$$\alpha' \left(1 - \frac{1}{t^T + 1}\right) > \alpha.$$

Set $M = (1-\delta)^{-1} t^q / \binom{q}{k}$ for notational convenience. We require of q_0 that $\alpha' M > t^T$ for all $q \geq q_0$. We now let $a \in [t^T, t^{T+1}) \cap \mathbb{N}$ and s be such that $at^s < \alpha' M \leq (a+1)t^s$. (These exist as the intervals $(at^s, (a+1)t^s]$ have union (t^T, ∞).) The upper bound on at^s gives the desired upper bound on $(1-\delta)at^s$. We also have a lower bound

$$at^s = \left(1 - \frac{1}{a+1}\right)(a+1)t^s \geq \left(1 - \frac{1}{t^T+1}\right)\alpha' M > \alpha M$$

so that $n \leq \alpha(1-\delta)M \leq (1-\delta)at^s$. □

3. Two-batch strategy.

We now fix a positive real $\alpha < (t/E)^k$. In this section we provide the two-batch strategy for Paul with a total number q of questions that works for any $n < \alpha \frac{t^q}{\binom{q}{k}}$, if q is sufficiently large.

To avoid trivialities we assume throughout the section that

$$n = \lfloor \alpha \frac{t^q}{\binom{q}{k}} \rfloor.$$

3.1. First batch.

In this subsection we give the strategy for Paul's first batch of questions and estimate the number of possibilities left after Carole provides her answers to these questions.

The strategy is as follows. Paul fixes α' with $\alpha < \alpha' < (t/E)^k$, then fixes a positive δ so that $\frac{\alpha'}{1-\delta} < (t/E)^k$. Finally, Paul fixes a, s as given by Lemma 2.2.

Remark 3.1. Because of the way it was chosen, $s = q - k \log_t q + O(1)$. In particular, $s \to \infty$ and $q - s \to \infty$ as $q \to \infty$.

Once these quantities are all fixed, Paul's first batch consists of $q_1 = s$ queries, and here is the strategy he employs for it.

Paul identifies the n possible answers with distinct pairs $(i, \mathbf{u})$, with $1 \leq i \leq a$ and $\mathbf{u} = u_1 \ldots u_s \in \mathcal{B}_{s,t}^r$. Here, for definiteness, we may fix $r = r(s) = \lfloor s^{2/3} \rfloor$, although we observe that any $r(s)$ such that $\sqrt{s} \ll r(s) \ll s$ would do. That there are sufficiently many such pairs follows from Lemmas 2.1 and 2.2.

Paul's first batch of queries is then simple to describe. For $1 \leq i \leq s$ he asks, "What is the value of u_i?"

Carole's first batch of responses gives a string $\mathbf{w} = w_1 \cdots w_s$. If Carole always responded truthfully, then $\mathbf{w}$ would necessarily be a balanced string. She is allowed to lie at most k times; hence $\mathbf{u}$ must be *nearly balanced* in the sense that every letter of $\mathcal{A}$ must appear in the response string $\mathbf{u}$ at most $\frac{s}{t} + r + k$ times.

Let $\mathbf{x} = (x_0, \ldots, x_k)$ denote the state of the position after these responses. That is, let x_j be the number of possibilities for which Carole has lied precisely j times.

LEMMA 3.2. *For each* $0 \leq j \leq k$, $x_j \leq a\frac{E^j}{j!}(\frac{s}{t} + r + k)^j$.

Proof. Let $\mathbf{w} = w_1 \cdots w_s$ be Carole's actual response. Then x_j is the number of $(i, \mathbf{w}')$ with $1 \leq i \leq a$ and where $\mathbf{w}, \mathbf{w}'$ differ in precisely j places and furthermore (and crucially) each such place is an allowable lie pattern. There are E^j sequences $(a_1, b_1), \ldots, (a_j, b_j)$ where the $a_i, b_i \in \mathcal{A}$ and (a_i, b_i) is an allowable lie pattern. For each such sequence there are at most $(\frac{s}{t} + r + k)^j$ sequences of positions $i_1, \ldots, i_j$ so that in Carole's response $\mathbf{w}$ the i_lth position had letter b_l. For each of these at most $E^j(\frac{s}{t} + r + k)^j$ possibilities Carole may have lied by changing the i_lth position from a_l. This gives every possible $\mathbf{w}'$. Each $\mathbf{w}'$ has been counted $j!$ times as you can permute the sequence $i_1, \ldots, i_j$ in that many ways. Thus the number of possible $\mathbf{w}'$ is at most $\frac{E^j}{j!}(\frac{s}{t} + r + k)^j$. Finally, there are a choices of i with $1 \leq i \leq a$. □

3.2. Packing is the same as winning. In this subsection we will provide the instruments to use for the second batch of questions, in the strategy we give for Paul.

A liar game in which all questions must be asked in a single batch of Q questions can be described as a packing problem. Let the alphabet $\mathcal{A}$ and channel C be fixed as given by Definition 1.

We use a notation from [5].

DEFINITION 5. *For any* $j \geq 0$ *and any* $\mathbf{w} = w_1 \cdots w_Q \in \mathcal{A}^Q$ *the* j*-shadow of* $\mathbf{w}$ *is the set of* $\mathbf{w}' = w_1' \cdots w_Q'$ *such that*

1. $w_i \neq w_i'$ *for at most* j *values* $1 \leq i \leq Q$.
2. *If* $w_i \neq w_i'$, *then* $(w_i, w_i') \in C$.

For the benefit of the reader, we have included the following example.

Example. Assume Paul, after a first batch of seven questions, has received the message 1443532 through the channel C of Figure 2, and suppose he knows that at most two errors have been made. The 2-shadow of $(1, 4, 4, 3, 5, 3, 2)$ is given by the set $A \cup B$, where A is the set of possibilities in case exactly one error was made and B is the set of possibilities if exactly two errors were made:

$$\begin{aligned} A = \{&(1,1,4,3,5,3,2), (1,4,1,3,5,3,2), (1,4,4,5,5,3,2), (1,4,4,3,5,5,2), \\ &(1,4,4,3,5,3,1), (1,4,4,3,5,3,4)\}, \end{aligned}$$

$$\begin{aligned} B = \{&(1,1,1,3,5,3,2), (1,1,4,5,5,3,2), (1,1,4,3,5,5,2), (1,1,4,3,5,3,1), \\ &(1,1,4,3,5,3,4), (1,4,1,5,5,3,2), (1,4,1,3,5,5,2), (1,4,1,3,5,3,1), \\ &(1,4,1,3,5,3,4), (1,4,4,5,5,5,2), (1,4,4,5,5,3,1), (1,4,4,5,5,3,4), \\ &(1,4,4,3,5,5,1), (1,4,4,3,5,5,4)\}. \end{aligned}$$

THEOREM 3.3. *Paul wins the Q question one-batch liar game from starting state $(x_0, \dots, x_k)$ if and only if there exist x_j $(k-j)$-shadows in $\mathcal{A}^Q$, all vertex disjoint for every $0 \le j \le k$.*

Remark 3.4. To clarify, we require even when $j \neq j'$ that no $(k-j)$-shadow overlaps any $(k-j')$-shadow.

Proof. Let $\mathbf{w}_l^j$, $0 \le j \le k$, $1 \le l \le x_j$, be such that the $(k-j)$-shadows of $\mathbf{w}_l^j$ are all vertex disjoint. Paul identifies the x_j possibilities for which Carole may lie $(k-j)$ times with $\mathbf{w}_l^j$. Paul asks for the coordinates of the vector. If the correct answer is $\mathbf{w}_l^j$, then Carole must respond with an element of its $(k-j)$-shadow. The disjointness of these shadows means that any response $\mathbf{w}^*$ of Carole is in precisely one such shadow and therefore Paul can determine which one.

We omit the proof in the other direction as we shall not be requiring it; it is essentially the same as the one for our "packing is equivalent to winning" argument of [5]. □

3.3. Second batch/endgame. Here we show that a simple greedy algorithm allows the packing Paul requires from Theorem 3.3, for the second batch of questions. Indeed, we show that Paul can win even if Carole's lies to the second batch of questions are unrestricted by the channel. The j-ball with center $\mathbf{w} \in \mathcal{A}^Q$ is the set of $\mathbf{w}' \in \mathcal{A}^Q$ that differ from $\mathbf{w}$ in at most j places. Let $F(Q,t,j)$ denote the size of the j-ball. Then

$$F(Q,t,j) = \sum_{l=0}^{j} \binom{Q}{l} (t-1)^l.$$

Note that the j-ball of $\mathbf{w}$ contains its j-shadow and is equal to its j-shadow when the channel C consists of all $(x,y) \in \mathcal{A} \times \mathcal{A}$. Note also that $F(Q,t,0) = 1$.

THEOREM 3.5. *Let $x_0, \dots, x_k$ satisfy*

$$\sum_{j=0}^{k} x_j F(Q,t,2(k-j)) \le t^Q.$$

Then there exist x_j $(k-j)$-balls in $\mathcal{A}^Q$ for $0 \le j \le k$, all $\sum_{j=0}^{k} x_j$ of them mutually disjoint.

Proof. We select the centers $\mathbf{w}_l^j$, $0 \le j \le k$, $1 \le l \le x_j$, sequentially. We do this in increasing order of j, first selecting the x_0 centers of k-balls, then the x_1 centers of $(k-1)$-balls, and continuing until finally selecting the x_k centers of 0-balls. We insist only that no new center selected lie in the $2(k–j)$-ball of any previously selected center $\mathbf{w}_l^j$. The assumed inequality gives that this prohibits less than t^Q vertices from being selected and therefore some $\mathbf{w} \in \mathcal{A}^Q$ is available. Consider any two centers selected in the order of their selection, say, $\mathbf{w}_l^j, \mathbf{w}'^{j'}_{l'}$. The ordering of selection ensures that $j \le j'$. As $\mathbf{w}^{j'}_{l'}$ does not lie in the $2(k-j)$-ball of $\mathbf{w}_l^j$, the $(k-j)$-balls of $\mathbf{w}_l^j, \mathbf{w}^{j'}_{l'}$ are disjoint and so the $(k–j)$-ball of $\mathbf{w}_l^j$ does not overlap the smaller (or equal) $(k–j')$-ball of $\mathbf{w}'^{j'}_{l'}$. □

To conclude the proof we need only show that $x_0, \dots, x_k$ satisfying the upper bounds of Lemma 3.2 will satisfy the conditions of Theorem 3.5 with $Q = q - s$. The most important case is x_k. We have $x_k \le a\frac{E^k}{k!}(\frac{s}{t} + r + k)^k$. We examine this asymptotically as q (and hence both s and $q-s$) approach infinity. As $q \sim s$, $r = o(s)$, and t, k are fixed, $(\frac{s}{t} + r + k)^k = (q/t)^k(1 + o(1))$ so that $x_k \le a(E/t)^k\binom{q}{k}(1 + o(1))$.

As $(1-\delta)at^s < \alpha' t^q / \binom{q}{k}$ we find

$$x_k \leq \frac{\alpha'}{1-\delta}(E/t)^k t^{q-s}(1+o(1)).$$

Paul's careful choice of δ sufficiently small ensures that we may express $x_k = (1-\Omega(1))t^{q-s}$.

For $0 \leq j < k$ we may use a more coarse upper bound for x_j, by absorbing $a, E^j, j!$ into the constant factor $x_j = O(s^j) = O(q^j)$. Furthermore, $n = \Theta(t^s)$ and $n = \Theta(t^q q^{-k})$ so $q^k = \Theta(t^{q-s})$, and therefore $x_j = O(t^{(q-s)j/k})$. (Note that x_j is bounded above by a *fractional power* of the number of elements in $\mathcal{A}^{q-s}$.) We bound $F(q-s, t, 2(k-j)) = O((q-s)^{2(k-j)})$ which is only polynomial in $q-s$. Hence

$$\sum_{j=0}^{k-1} x_j F(Q, t, 2(k-j)) = \sum_{j=0}^{k-1} O\left(t^{(q-s)j/k}\ (q-s)^{2(k-j)}\right) = o(t^{q-s}).$$

That is, the $x_0, \ldots, x_{k-1}$ terms (corresponding to the cases in which Carole did not use the maximal permissible number of lies in her responses to the first batch of questions) are asymptotically negligible and

$$\sum_{j=0}^{k} x_j F(Q, t, 2(k-j)) = (1-\Omega(1))t^{q-s}.$$

For q sufficiently large, $q-s$ is therefore sufficiently large so that the conditions of Theorem 3.5 hold. For such large q Paul therefore has a second batch of $(q-s)$ questions that allows him to determine Carole's answer.

4. Conclusions. In this paper we have proved that there exists a strategy for Paul which allows him to win the (n, k, C) two-batch liar game with q questions for $n \sim \left(\frac{t}{E}\right)^k \frac{t^q}{\binom{q}{k}}$. We have done this by giving a strategy that allows him to ask *most* of the questions in the first offline batch ($q_1 \sim q$) and use an exponentially smaller number of questions ($q_2 \sim k \log q$) for the second batch.

One may argue that this strategy is a very desirable one, since it allows for most of the questions to be asked in an offline fashion at first and uses only an exponentially smaller number of "corrective" questions in the second batch. At the same time, one might also argue that Paul receives much feedback from Carole's answers to his first batch of questions, and that is why the number of corrective questions needs to be much smaller.

We raise two open questions. First, to what extent can the results of our work be tightened? We note that in [9] second order terms were given for the original Rényi–Ulam game, and perhaps similar results apply in our more general setting. Second, suppose Paul does not have the freedom of choosing the size q_1 of his first batch of questions. For what range of values q_1 can Paul still win?

We close with the connection between the two-batch problem and the one-batch, or completely offline, problem. The asymptotics of $A^{-}_{C,k}(q)$ (the maximal n for the one-batch variant of the liar game) remain open. Indeed, this has been a prime motivating force in our research. Is $A^{-}_{C,k}(q) \sim A_{C,k}(q)$? In words, do the completely offline and completely online problems have the same asymptotic solution? We feel (mildly) that our results point in this direction. We hope that the tools we constructed for our analysis will be helpful in extending the asymptotics to these cases.

Acknowledgment. Ioana Dumitriu would like to thank the Miller Institute for Basic Research in Sciences for their support.

REFERENCES

[1] N. Alon and J. Spencer, *The Probabilistic Method*, 2nd ed., John Wiley, New York, 2000.

[2] E. R. Berlekamp, *Block Coding for the Binary Symmetric Channel with Noiseless, Delayless Feedback*, in Error-Correcting Codes, H.B. Mann, ed., John Wiley, New York, 1968, pp. 61–88.

[3] F. Cicalese and D. Mundici, *Optimal coding with one asymmetric error: Below the sphere packing bound*, in Proceedings of the 6th Annual International Conference on Computing and Combinatorics–COCOON'2000, Lecture Notes in Comput. Sci., 1858, Springer-Verlag, 2000, pp. 159–169.

[4] I. Dumitriu and J. Spencer, *A halfliar's game*, *Theoret. Comput. Sci.*, 313 (2004), pp. 353–369.

[5] I. Dumitriu and J. Spencer, *The Liar Game over an Arbitrary Channel*, Combinatorica, 25 (2005), pp. 537–559.

[6] A. Pelc, *Searching games with errors—fifty years of coping with liars*, Theoret. Comput. Sci., 270 (2002), pp. 71–109.

[7] A. Rényi, *A Diary on Information Theory*, John Wiley, New York, 1984.

[8] J. Spencer, *Ulam's searching problem with a fixed number of lies*, Theoret. Comput. Sci., 95 (1992), pp. 307–321.

[9] J. Spencer and C. Yan, *The halflie problem*, J. Combinat. Theory Ser. A, 103 (2003), pp. 69–89.

[10] S. M. Ulam, *Adventures of a Mathematician*, Scribners, New York, 1976.

SIAM J. DISCRETE MATH.
Vol. 19, No. 4, pp. 1065–1073

BISUBMODULAR FUNCTION MINIMIZATION[*]

SATORU FUJISHIGE[†] AND SATORU IWATA[‡]

Abstract. This paper presents the first combinatorial polynomial algorithm for minimizing bisubmodular functions, extending the scaling algorithm for submodular function minimization due to Iwata, Fleischer, and Fujishige. Since the rank functions of delta-matroids are bisubmodular, the scaling algorithm naturally leads to the first combinatorial polynomial algorithm for testing membership in delta-matroid polyhedra.

Key words. bisubmodular function, delta-matroid, scaling algorithm

AMS subject classification. 90C27

DOI. 10.1137/S0895480103426339

1. Introduction.

Let V be a finite nonempty set of cardinality n and 3^V denote the set of ordered pairs of disjoint subsets of V. Two binary operations $\sqcup$ and $\sqcap$ on 3^V are defined by

$$(X_1, Y_1) \sqcup (X_2, Y_2) = ((X_1 \cup X_2) \backslash (Y_1 \cup Y_2), (Y_1 \cup Y_2) \backslash (X_1 \cup X_2)),$$
$$(X_1, Y_1) \sqcap (X_2, Y_2) = (X_1 \cap X_2, Y_1 \cap Y_2).$$

A function $f : 3^V \to \mathbf{R}$ is called *bisubmodular* if it satisfies

$$f(X_1, Y_1) + f(X_2, Y_2) \geq f((X_1, Y_1) \sqcup (X_2, Y_2)) + f((X_1, Y_1) \sqcap (X_2, Y_2))$$

for any (X_1, Y_1) and (X_2, Y_2) in 3^V. This paper presents the first combinatorial polynomial algorithm for minimizing bisubmodular functions.

Examples of bisubmodular functions include the rank functions of delta-matroids introduced independently by Bouchet [3] and Chandrasekaran–Kabadi [6]. A delta-matroid is a set system $(V, \mathcal{F})$ with $\mathcal{F}$ being a nonempty family of subsets of V that satisfies the following exchange property:

$$\forall F_1, F_2 \in \mathcal{F}, \forall v \in F_1 \triangle F_2, \exists u \in F_1 \triangle F_2 : F_1 \triangle \{u, v\} \in \mathcal{F},$$

where $\triangle$ denotes the symmetric difference. A slightly restricted set system with an additional condition $\emptyset \in \mathcal{F}$ had been introduced by Dress–Havel [11]. A member of $\mathcal{F}$ is called a feasible set of the delta-matroid. Note that the base and the independent-set families of a matroid satisfy this exchange property. Thus, a delta-matroid is a generalization of a matroid.

Chandrasekaran–Kabadi [6] showed that the rank function $\varrho : 3^V \to \mathbf{Z}$ defined by

$$\varrho(X, Y) = \max\{|X \cap F| - |Y \cap F| \mid F \in \mathcal{F}\}$$

[*] Received by the editors April 25, 2003; accepted for publication (in revised form) September 22, 2005; published electronically February 3, 2006. A preliminary version of this paper appeared in *Proceedings of the Eighth Conference on Integer Programming and Combinatorial Optimization*, LNCS 2081, Springer-Verlag, Berlin, 2001, pp. 160–169.

http://www.siam.org/journals/sidma/19-4/42633.html

[†] Research Institute for Mathematical Sciences, Kyoto University, Kyoto 606-8502, Japan (fujishig@kurims.kyoto-u.ac.jp).

[‡] Department of Mathematical Informatics, University of Tokyo, Tokyo 113-8656, Japan (iwata@mist.i.u-tokyo.ac.jp).

is bisubmodular. The convex hull of the characteristic vectors of the feasible sets is described by

$$\mathrm{P}(\varrho) = \{x \mid x \in \mathbf{R}^V,\, \forall (X,Y) \in 3^V : x(X) - x(Y) \le \varrho(X,Y)\},$$

which is called the delta-matroid polyhedron. This fact follows from the greedy algorithm [3, 6] for optimizing a linear function over the feasible sets.

Given a vector $x \in \mathbf{R}^V$, one can test if x belongs to $\mathrm{P}(\varrho)$ by minimizing a bisubmodular function $f(X,Y) = \varrho(X,Y) - x(X) + x(Y)$. Even for such a special case of bisubmodular function minimization, no combinatorial algorithm was known to run in polynomial time. This is in contrast with the matroid polyhedron, for which Cunningham [7] devised a combinatorial strongly polynomial algorithm for testing membership.

A simple example of a delta-matroid is a matching delta-matroid [4], whose feasible sets are the perfectly matchable vertex subsets of an undirected graph. The corresponding delta-matroid polyhedron is the matchable set polytope [2]. For this special case, Cunningham–Green-Krótki [10] developed an augmenting path algorithm for solving the separation problem in polynomial time with the aid of the scaling technique.

A bisubmodular function also generalizes a submodular (set) function. Let 2^V denote the family of all the subsets of V. A function $g : 2^V \to \mathbf{R}$ is called *submodular* if it satisfies

$$g(Z_1) + g(Z_2) \ge g(Z_1 \cup Z_2) + g(Z_1 \cap Z_2)$$

for any $Z_1, Z_2 \subseteq V$. For a submodular function g, we define a bisubmodular function $f : 3^V \to \mathbf{R}$ by

$$f(X,Y) = g(X) + g(V \backslash Y) - g(V).$$

If (X,Y) is a minimizer of f, then both X and $V \backslash Y$ are minimizers of g. Thus, bisubmodular function minimization (BSFM) is a generalization of submodular function minimization.

The first polynomial algorithm for submodular function minimization is given by Grötschel–Lovász–Schrijver [16]. They also give the first strongly polynomial algorithms in [17]. Their algorithms rely on the ellipsoid method, which is not efficient in practice. Two combinatorial strongly polynomial algorithms have been devised independently by Schrijver [23] and Iwata–Fleischer–Fujishige [18]. Both of these new algorithms are based on the combinatorial pseudopolynomial algorithm given by Cunningham [8]. The algorithm of Schrijver [23] directly achieves a strongly polynomial bound, whereas Iwata–Fleischer–Fujishige [18] develop a scaling algorithm with weakly polynomial time complexity and then convert it to a strongly polynomial one.

In the present paper, we extend the scaling algorithm of Iwata–Fleischer–Fujishige [18] to solve the minimization problem for integer-valued bisubmodular functions. The resulting algorithm runs in $\mathrm{O}(n^5 \log M)$ time, where M designates the maximum value of f. This bound is weakly polynomial. A strongly polynomial version will be presented in a forthcoming paper by McCormick–Fujishige [21].

As a generalization of the delta-matroid polyhedron, a *bisubmodular polyhedron*

$$\mathrm{P}(f) = \{x \mid x \in \mathbf{R}^V,\, \forall (X,Y) \in 3^V : x(X) - x(Y) \le f(X,Y)\}$$

is associated with a general bisubmodular function $f : 3^V \to \mathbf{R}$, where we assume $f(\emptyset,\emptyset) = 0$. The linear optimization problem over the bisubmodular polyhedron can be solved by the greedy algorithm of Dunstan–Welsh [12]. Based on the validity of this greedy algorithm, Qi [22] established a connection between bisubmodular functions and their convex extensions. This generalizes a result of Lovász [19] on submodular functions. Qi [22] also mentioned that the connection provides a polynomial algorithm for bisubmodular function minimization using the ellipsoid method. In contrast, our combinatorial algorithm directly deals with the separation problem for bisubmodular polyhedra.

The concept of delta-matroid is extended to that of jump system by Bouchet–Cunningham [5]. A jump system is a set of lattice points satisfying a certain axiom. Examples include the set of degree sequences of a graph [9]. Lovász [20] investigated the membership problem in jump systems and proved a min-max theorem for a fairly wide class of jump systems. The lattice points contained in an integral bisubmodular polyhedron form a jump system, called a convex jump system, and conversely the convex hull of a jump system is an integral bisubmodular polyhedron. An example of a convex jump system is the so-called b-matching degree sequence polyhedron [9]. A very recent paper of Zhang [24] presented an augmenting path separation algorithm for this polyhedron. The present paper provides a more general algorithmic approach to the membership problem in convex jump systems.

2. Bisubmodular polyhedra.

This section provides a preliminary on bisubmodular polyhedra. See [14, section 3.5 (b)] for more detail and background.

For any vector $y \in \mathbf{R}^V$, we denote $\|y\| = \sum_{v \in V} |y(v)|$. Concerning the shortest distance from a given vector x° to the bisubmodular polyhedron $\mathrm{P}(f)$, we have the following min-max relation, which is essentially equivalent to an earlier result of Cunningham–Green-Krótki [9].

Theorem 2.1 (Fujishige [15]). *For any bisubmodular function $f : 3^V \to \mathbf{R}$ and any vector $x^\circ \in \mathbf{R}^V$,*

$$\min\{\|x - x^\circ\| \mid x \in \mathrm{P}(f)\} = \max\{x^\circ(X) - x^\circ(Y) - f(X,Y) \mid (X,Y) \in 3^V\}$$

holds. Moreover, if f and x° are integer valued, then there is an integral vector x that attains the minimum in the left-hand side.

When $x^\circ = \mathbf{0}$, the min-max relation characterizes the minimum value of f. Our combinatorial algorithm is built on this characterization.

We now turn to the greedy algorithm for computing an extreme point of a bisubmodular polyhedron. See also [1] for related structural results on extreme points of bisubmodular polyhedra.

Let $\sigma : V \to \{+,-\}$ be a sign function. For any subset $U \subseteq V$, we denote by $U|\sigma$ the pair $(X,Y) \in 3^V$ with $X = \{u \mid u \in U, \sigma(u) = +\}$ and $Y = \{u \mid u \in U, \sigma(u) = -\}$. We also write $f(U|\sigma) = f(X,Y)$ for any function $f : 3^V \to \mathbf{R}$, and $x(U|\sigma) = x(X) - x(Y)$ for any vector $x \in \mathbf{R}^V$.

Let $L = (v_1, \cdots, v_n)$ be a linear ordering of V. For each $j = 1, \ldots, n$, let $L(v_j) = \{v_1, \ldots, v_j\}$. The *greedy algorithm* with respect to L and a sign function σ assigns $y(v) := \sigma(v)\{f(L(v)|\sigma) - f(L(v)\backslash\{v\}|\sigma)\}$ for each $v \in V$. Then the resulting vector $y \in \mathbf{R}^V$ is an extreme point of the bisubmodular polyhedron $\mathrm{P}(f)$.

Given a weight function $w : V \to \mathbf{R}$, construct a linear ordering $L = (v_1, \ldots, v_n)$ and a sign function σ that satisfies $|w(v_1)| \geq \cdots \geq |w(v_n)|$ and $w(v) = \sigma(v)|w(v)|$ for each $v \in V$. Then the vector y generated by the greedy algorithm with respect to L and σ maximizes the linear function $\sum_{v \in V} w(v)y(v)$ over the bisubmodular polyhedron $\mathrm{P}(f)$.

3. Scaling algorithm. This section presents a scaling algorithm for minimizing an integer-valued bisubmodular function $f : 3^V \to \mathbf{Z}$, provided that an oracle for evaluating the function value is available.

The scaling algorithm works with a positive parameter δ. The algorithm keeps a vector $x \in \mathrm{P}(f)$ as a convex combination of extreme points of $\mathrm{P}(f)$ indexed by finite set I. Namely, $x = \sum_{i \in I} \lambda_i y_i$ with $\lambda_i > 0$ for each $i \in I$ and $\sum_{i \in I} \lambda_i = 1$. Each extreme point y_i is generated by the greedy algorithm with respect to L_i and σ_i. It also keeps a pair of functions $\varphi : V \times V \to \mathbf{R}$ and $\psi : V \times V \to \mathbf{R}$. The function φ is skew-symmetric, i.e., $\varphi(u, v) + \varphi(v, u) = 0$ for any $u, v \in V$, while ψ is symmetric, i.e., $\psi(u, v) = \psi(v, u)$ for any $u, v \in V$. These functions are called *δ-feasible* if they satisfy $-\delta \leq \varphi(u, v) \leq \delta$ and $-\delta \leq \psi(u, v) \leq \delta$ for any $u, v \in V$. The boundaries $\partial\varphi$ and $\partial\psi$ are defined by $\partial\varphi(u) = \sum_{v \in V} \varphi(u, v)$ and $\partial\psi(u) = \sum_{v \in V} \psi(u, v)$.

The algorithm starts with an extreme point $x \in \mathrm{P}(f)$ generated by the greedy algorithm with respect to a linear ordering L and a sign function σ. The initial value of δ is given by $\delta := \|x\|/n^2$, and the initial values of φ and ψ are zero.

Each scaling phase starts by cutting the value of δ in half. Then it modifies φ and ψ to make them δ-feasible. This can be done by setting each $\varphi(u, v)$ and $\psi(u, v)$ to the closest values in the interval $[-\delta, \delta]$.

The rest of the scaling phase aims at decreasing $\|z\|$ for $z = x + \partial\varphi + \partial\psi$. It uses three procedures: Augment, Double-Exchange, and Tail-Exchange. Procedure Augment decreases $\|z\|$ by δ, whereas Double-Exchange and Tail-Exchange modify x, φ, ψ keeping z invariant. The scaling phase terminates when none of these procedures are applicable. Then $\|z\|$ is shown to be small enough to keep moderate the number of applications of Augment in the next scaling phase. Furthermore, if $\delta < 1/3n^2$, then the algorithm detects a pair (X, Y) that minimizes f.

Given δ-feasible φ and ψ, the algorithm constructs an auxiliary directed graph $G(\varphi, \psi)$ as follows. Let V^+ and V^- be two copies of V. For each $v \in V$, we denote its copies by $v^+ \in V^+$ and $v^- \in V^-$. The vertex set of $G(\varphi, \psi)$ is $V^+ \cup V^-$. For any subset $U \subseteq V$, define $U^+ = \{u^+ \mid u \in U\}$ and $U^- = \{u^- \mid u \in U\}$. The arc set $A(\varphi, \psi) = A(\varphi) \cup A(\psi)$ of $G(\varphi, \psi)$ is defined by

$$A(\varphi) = \{(u^+, v^+) \mid u \neq v,\, \varphi(u, v) \leq 0\} \cup \{(u^-, v^-) \mid u \neq v,\, \varphi(u, v) \geq 0\},$$
$$A(\psi) = \{(u^+, v^-) \mid \psi(u, v) \leq 0\} \cup \{(u^-, v^+) \mid \psi(u, v) \geq 0\}.$$

Note that, for any $u, v \in V$ and any sign function η on V, the graph $G(\varphi, \psi)$ has a directed path from $u^{\eta(u)}$ to $v^{\eta(v)}$ if and only if $G(\varphi, \psi)$ has a directed path from $v^{-\eta(v)}$ to $u^{-\eta(u)}$.

Let $S = \{v \mid v \in V, z(v) \leq -\delta\}$ and $T = \{v \mid v \in V, z(v) \geq \delta\}$. A simple directed path in $G(\varphi, \psi)$ from $S^+ \cup T^-$ to $S^- \cup T^+$ is called a *δ-augmenting path.* If there exists a δ-augmenting path P, the algorithm applies the following δ-augmentation to φ and ψ.

Augment$(\delta, P, \varphi, \psi)$:

- For each (u^+, v^+) in P, $\varphi(u, v) := \varphi(u, v) + \delta/2$ and $\varphi(v, u) := \varphi(v, u) - \delta/2$.
- For each (u^-, v^-) in P, $\varphi(u, v) := \varphi(u, v) - \delta/2$ and $\varphi(v, u) := \varphi(v, u) + \delta/2$.
- For each (u^+, v^-) in P, $\psi(u, v) := \psi(u, v) + \delta/2$ and $\psi(v, u) := \psi(v, u) + \delta/2$.
- For each (u^-, v^+) in P, $\psi(u, v) := \psi(u, v) - \delta/2$ and $\psi(v, u) := \psi(v, u) - \delta/2$.

Note that a δ-augmentation does not change x, maintains δ-feasibility, and decreases $\|z\|$ by δ.

After each δ-augmentation, the algorithm computes an expression of x as a convex combination of affinely independent extreme points of $\mathrm{P}(f)$ chosen from among $\{y_i \mid$

```
Double-Exchange$(i, u, v)$;
$\beta := f(L_i(u)\backslash\{v\}|\sigma_i) - f(L_i(u)|\sigma_i) + \sigma_i(v)y_i(v)$;
$\alpha := \min\{\delta,\, \lambda_i\beta\}$;
If $\alpha < \lambda_i\beta$ then
        $k \leftarrow$ a new index;
        $I := I \cup \{k\}$;
        $\lambda_k := \lambda_i - \alpha/\beta$;
        $\lambda_i := \alpha/\beta$;
        $y_k := y_i$;
        $L_k := L_i$;
Update $L_i$ by interchanging $u$ and $v$;
$y_i := y_i + \beta(\sigma_i(u)\chi_u - \sigma_i(v)\chi_v)$;
$x := x + \alpha(\sigma_i(u)\chi_u - \sigma_i(v)\chi_v)$;
If $\sigma_i(u) = \sigma_i(v)$ then
        $\varphi(u, v) := \varphi(u, v) - \sigma_i(u)\alpha$;
        $\varphi(v, u) := \varphi(v, u) + \sigma_i(u)\alpha$;
Else
        $\psi(u, v) := \psi(u, v) - \sigma_i(u)\alpha$;
        $\psi(v, u) := \psi(v, u) - \sigma_i(u)\alpha$.
```

FIG. 3.1. *Algorithmic description of Procedure* Double-Exchange(i, u, v).

$i \in I\}$. This can be done by a standard linear programming technique using Gaussian elimination.

If there is no δ-augmenting path, let $X^+ \subseteq V^+$ and $Y^- \subseteq V^-$ be the sets of vertices reachable by directed paths from $S^+ \cup T^-$. Then we have $S \subseteq X$, $T \subseteq Y$, and $X \cap Y = \emptyset$. For each $i \in I$, consider a pair of disjoint subsets $W_i = \{u \mid u^{\sigma_i(u)} \in X^+ \cup Y^-\}$ and $R_i = \{u \mid u^{\sigma_i(u)} \in X^- \cup Y^+\}$. Note that $W_i \cup R_i = X \cup Y$. We now introduce two procedures: Double-Exchange and Tail-Exchange.

Procedure Double-Exchange(i, u, v) is applicable if u immediately succeeds v in L_i and either $u \in W_i$ and $v \notin W_i$ or $u \notin R_i$ and $v \in R_i$ hold. Such a triple (i, u, v) is called *active*. The first step of the procedure is to compute

$$\beta := f(L_i(u)\backslash\{v\}|\sigma_i) - f(L_i(u)|\sigma_i) + \sigma_i(v)y_i(v).$$

Then it interchanges u and v in L_i and updates y_i as $y_i := y_i + \beta(\sigma_i(u)\chi_u - \sigma_i(v)\chi_v)$. The resulting y_i is an extreme point generated by the new linear ordering L_i and sign function σ_i.

If $\lambda_i\beta \leq \delta$, Double-Exchange$(i, u, v)$ is called *saturating*. Otherwise, it is called *nonsaturating*. In the nonsaturating case, the procedure adds to I a new index k with y_k, σ_k and L_k being the previous y_i, σ_i and L_i, and assigns $\lambda_k := \lambda_i - \delta/\beta$ and $\lambda_i := \delta/\beta$. In both cases, x moves to $x := x + \alpha(\sigma_i(u)\chi_u - \sigma_i(v)\chi_v)$ with $\alpha = \min\{\delta, \lambda_i\beta\}$. In order to keep z invariant, the procedure finally modifies φ or ψ appropriately. If $\sigma_i(u) = \sigma_i(v)$, it updates $\varphi(u, v) := \varphi(u, v) - \sigma_i(u)\alpha$ and $\varphi(v, u) := \varphi(v, u) + \sigma_i(u)\alpha$. On the other hand, if $\sigma_i(u) \neq \sigma_i(v)$, then $\psi(u, v) := \psi(u, v) - \sigma_i(u)\alpha$ and $\psi(v, u) := \psi(v, u) - \sigma_i(u)\alpha$. A formal description of Double-Exchange is given in Figure 3.1.

LEMMA 3.1. *As a result of nonsaturating* Double-Exchange(i, u, v)*, a new vertex joins $X \cup Y$ or a δ-augmenting path appears in $G(\varphi, \psi)$.*

```
Tail-Exchange$(i, v)$;
$\sigma_i(v) := -\sigma_i(v)$;
$\beta := f(V|\sigma_i) - f(V\backslash\{v\}|\sigma_i) - \sigma_i(v)y_i(v)$;
$\alpha := \min\{\delta,\ \lambda_i\beta\}$;
If $\alpha < \lambda_i\beta$ then
    $k \leftarrow$ a new index;
    $I := I \cup \{k\}$;
    $\lambda_k := \lambda_i - \alpha/\beta$;
    $\lambda_i := \alpha/\beta$;
    $y_k := y_i$;
    $L_k := L_i$;
$y_i := y_i + \sigma_i(v)\beta\chi_v$;
$x := x + \sigma_i(v)\alpha\chi_v$;
$\psi(v, v) := \psi(v, v) - \sigma_i(v)\alpha$.
```

FIG. 3.2. *Algorithmic description of Procedure* Tail-Exchange(i, v).

Proof. The nonsaturating Double-Exchange(i, u, v) updates φ or ψ so that a new arc $(u^{\sigma_i(u)}, v^{\sigma_i(v)})$ should appear in $G(\varphi, \psi)$. Recall that either $u \in W_i$ and $v \notin W_i$ or $u \notin R_i$ and $v \in R_i$ holds. If $u \in W_i$ and $v \notin R_i \cup W_i$, the new arc makes $v^{\sigma_i(v)}$ reachable from $S^+ \cup T^-$. If $v \in R_i$ and $u \notin W_i \cup R_i$, the new arc makes $u^{\sigma_i(u)}$ reachable to $S^- \cup T^+$. Thus, in these cases, a new vertex v or u is added to $X \cup Y$. Finally, if $u \in W_i$ and $v \in R_i$, the new arc yields a δ-augmenting path. □

Procedure Tail-Exchange(i, v) is applicable if v is the last element in L_i and $v \in R_i$. Such a pair (i, v) is also called *active*. The first step of the procedure is to reverse the sign $\sigma_i(v)$. It then computes

$$\beta := f(V|\sigma_i) - f(V\backslash\{v\}|\sigma_i) - \sigma_i(v)y_i(v)$$

and updates $y_i := y_i + \sigma_i(v)\beta\chi_v$. The resulting y_i is an extreme point generated by L_i and the new σ_i.

If $\lambda_i\beta \leq \delta$, Tail-Exchange$(i, v)$ is called *saturating*. Otherwise, it is called *nonsaturating*. In the nonsaturating case, the procedure adds to I a new index k with y_k, σ_k and L_k being the previous y_i, σ_i and L_i, and assigns $\lambda_k := \lambda_i - \delta/\beta$ and $\lambda_i := \delta/\beta$. In both cases, x moves to $x := x + \sigma_i(v)\alpha\chi_v$ with $\alpha = \min\{\delta, \lambda_i\beta\}$. In order to keep z invariant, the procedure finally modifies ψ as $\psi(v, v) := \psi(v, v) - \sigma_i(v)\alpha$. A formal description of Tail-Exchange is given in Figure 3.2.

LEMMA 3.2. *As a result of nonsaturating* Tail-Exchange(i, v), *a δ-augmenting path appears in $G(\varphi, \psi)$.*

Proof. Suppose the algorithm applies Tail-Exchange(i, v) with $\sigma_i(v) = \tau$. Then $v^{-\tau}$ is reachable from $S^+ \cup T^-$ and v^τ is reachable to $S^- \cup T^+$ in $G(\varphi, \psi)$. The nonsaturating Tail-Exchange(i, v) changes $\sigma_i(v)$ to $-\tau$ and updates $\psi(v, v)$ so that a new arc $(v^{-\tau}, v^\tau)$ appears in $G(\varphi, \psi)$, which yields a δ-augmenting path from $S^+ \cup T^-$ to $S^- \cup T^+$. □

If there is no δ-augmenting path and neither Double-Exchange nor Tail-Exchange is applicable, the algorithm terminates the scaling phase. Then it goes to the next scaling phase unless $\delta < 1/3n^2$. If $\delta < 1/3n^2$, the algorithm terminates by returning the current (X, Y) as a minimizer of f.

A formal description of our scaling algorithm BSFM is now given in Figure 3.3.

```
BSFM($f$):
Initialization:
    $L \leftarrow$ a linear ordering on $V$;
    $\sigma \leftarrow$ a sign function on $V$;
    $x \leftarrow$ an extreme vector in $\mathrm{P}(f)$ generated by $L$ and $\sigma$;
    $I := \{\ell\}$, $y_\ell := x$, $\lambda_\ell := 1$, $L_\ell := L$;
    $\varphi := \mathbf{0}$, $\psi := \mathbf{0}$;
    $\delta \leftarrow \|x\|/n^2$;
While $\delta \geq 1/3n^2$ do
    $\delta := \delta/2$;
    For $(u,v) \in V \times V$ do
        Change $\varphi(u,v)$ and $\psi(u,v)$ to the closest values in the interval $[-\delta, \delta]$;
    $S := \{v \mid x(v) + \partial\varphi(v) + \partial\psi(v) \leq -\delta\}$;
    $T := \{v \mid x(v) + \partial\varphi(v) + \partial\psi(v) \geq \delta\}$;
    $X^+ \leftarrow$ the set of vertices in $V^+$ reachable from $S^+ \cup T^-$ in $G(\varphi,\psi)$;
    $Y^- \leftarrow$ the set of vertices in $V^-$ reachable from $S^+ \cup T^-$ in $G(\varphi,\psi)$;
    $Q \leftarrow$ the set of active triples and active pairs;
    While $\exists\delta$-augmenting path or $Q \neq \emptyset$ do
        If $\exists P$: $\delta$-augmenting path then
            Augment($\delta, P, \varphi, \psi$);
            Update $S$, $T$, $X^+$, $Y^-$, $Q$;
            Express $x$ as $x = \sum_{i \in I} \lambda_i y_i$ by possibly smaller affinely independent
                subset $I$ and positive coefficients $\lambda_i > 0$ for $i \in I$;
        Else
            While $\nexists\delta$-augmenting path and $Q \neq \emptyset$ do
                Find an active $(i,u,v) \in Q$ or active $(i,v) \in Q$;
                Apply Double-Exchange($i,u,v$) or Tail-Exchange($i,v$);
                Update $X^+$, $Y^-$, $Q$;
Return $(X,Y)$;
End.
```

FIG. 3.3. *A scaling algorithm for bisubmodular function minimization.*

4. Validity and complexity.

This section is devoted to the analysis of our scaling algorithm. We first discuss the validity.

LEMMA 4.1. *At the end of each scaling phase, the current $(X,Y) \in 3^V$ and $z = x + \partial\varphi + \partial\psi$ satisfy $\|z\| \leq 2n\delta - f(X,Y)$.*

Proof. At the end of each scaling phase, we have $y_i(X) - y_i(Y) = f(X,Y)$ for each $i \in I$. Hence, x satisfies $x(X) - x(Y) = f(X,Y)$. By the definition of (X,Y), we have $\varphi(u,v) > 0$ for $u \in X$, $v \in V \backslash X$ and $\varphi(u,v) < 0$ for $u \in Y$, $v \in V\backslash Y$. These inequalities imply $\partial\varphi(X) = \sum\{\varphi(u,v) \mid u \in X, v \in V\backslash X\} > 0$ and $\partial\varphi(Y) = \sum\{\varphi(u,v) \mid u \in Y, v \in V\backslash Y\} < 0$. Similarly, we have $\psi(u,v) > 0$ for $u \in X$, $v \in V\backslash Y$ and $\psi(u,v) < 0$ for $u \in Y$, $v \in V\backslash X$, which imply $\partial\psi(X) = \sum\{\psi(u,v) \mid u \in X, v \in V\} > \theta$ and $\partial\psi(Y) = \sum\{\psi(u,v) \mid u \in Y, v \in V\} < \theta$, where $\theta = \sum\{\psi(u,v) \mid u \in X, v \in Y\}$. Since $S \subseteq X$ and $T \subseteq Y$, we have $z(v) \geq -\delta$ for $v \in V\backslash X$ and $z(v) \leq \delta$ for $v \in V\backslash Y$. Therefore, we have $\|z\| \leq -z(X) + z(Y) + 2n\delta \leq -x(X) + x(Y) + 2n\delta = -f(X,Y) + 2n\delta$. □

THEOREM 4.2. *The algorithm obtains a minimizer of f at the end of the last scaling phase.*

Proof. Note that $\delta < 1/3n^2$ at the end of the last scaling phase. For each $v \in V$, since $|\partial\varphi(v)| \leq (n-1)\delta$ and $|\partial\psi(v)| \leq n\delta$, we have $|x(v)| \leq |z(v)| + |\partial\varphi(v)| + |\partial\psi(v)| \leq |z(v)| + (2n-1)\delta$. Then it follows from Lemma 4.1 that $\|x\| \leq (2n^2+n)\delta - f(X,Y) < 1 - f(X,Y)$. For any $(X', Y') \in 3^V$, we have $f(X', Y') \geq x(X') - x(Y') \geq -\|x\| > f(X,Y) - 1$. Hence (X,Y) is a minimizer of the integer-valued function f. □

We now give a running time bound of our algorithm.

LEMMA 4.3. *Each scaling phase performs* $\mathrm{O}(n^2)$ *augmentations.*

Proof. At the beginning of each scaling phase, the algorithm modifies φ and ψ to make them δ-feasible (for the new δ). This changes $\|z\|$ by at most $2n^2\delta$. Therefore, by Lemma 4.1, the pair (X,Y) obtained at the end of the previous scaling phase must satisfy $\|z\| \leq 2n^2\delta + 4n\delta - f(X,Y)$ after updating φ and ψ at the beginning of the current scaling phase. On the other hand, at the end of the current scaling phase, we have $\|z\| \geq -z(X) + z(Y) \geq -x(X) + x(Y) - 2n^2\delta \geq -f(X,Y) - 2n^2\delta$. Thus, during the scaling phase, $\|z\|$ decreases by at most $4n\delta + 4n^2\delta$. Since each δ-augmentation decreases $\|z\|$ by δ, the number of δ-augmentations in the scaling phase is at most $4n^2 + 4n$, which is $\mathrm{O}(n^2)$. □

LEMMA 4.4. *The algorithm performs Procedure* Double-Exchange $\mathrm{O}(n^3)$ *times and* Tail-Exchange $\mathrm{O}(n^2)$ *times between δ-augmentations.*

Proof. In Double-Exchange, a vertex in W_i moves in L_i ahead of some vertex not in W_i and/or a vertex in R_i moves behind some vertex not in R_i. Procedure Tail-Exchange changes a vertex of R_i into W_i. No vertex goes out of W_i. A vertex of R_i can be switched to W_i by Tail-Exchange. However, it does not go out of $R_i \cup W_i$. Thus, for each $i \in I$, after at most $\mathrm{O}(n^2)$ applications of Double-Exchange and $\mathrm{O}(n)$ applications of Tail-Exchange to $i \in I$, the subset R_i is empty and $W_i = L_i(w)$ holds for some $w \in V$. At this point, neither Double-Exchange nor Tail-Exchange is applicable to $i \in I$.

After each δ-augmentation, the algorithm updates the convex combination $x = \sum_{i\in I} \lambda_i y_i$ so that $|I| \leq n+1$. A new index is added to I as a result of nonsaturating Double-Exchange(i,u,v) and Tail-Exchange(i,v). It follows from Lemmas 3.1 and 3.2 that this can happen at most $n-1$ times before the algorithm finds a δ-augmenting path or finishes the scaling phase. Hence, $|I|$ is always $\mathrm{O}(n)$, and the algorithm performs Double-Exchange $\mathrm{O}(n^3)$ times and Tail-Exchange $\mathrm{O}(n^2)$ times between δ-augmentations. □

Let M be the maximum value of f. Since $f(\emptyset,\emptyset) = 0$, the maximum value M is nonnegative.

THEOREM 4.5. *The scaling algorithm finds a minimizer of f in* $\mathrm{O}(n^5 \log M)$ *time.*

Proof. For the initial $x \in \mathrm{P}(f)$, let $B = \{v \mid x(v) > 0\}$ and $C = \{v \mid x(v) < 0\}$. Then we have $\|x\| = x(B) - x(C) \leq f(B,C) \leq M$. Hence the algorithm performs $\mathrm{O}(\log M)$ scaling phases. It follows from Lemmas 4.3 and 4.4 that each scaling phase performs $\mathrm{O}(n^5)$ function evaluations and arithmetic operations. Therefore the total running time is $\mathrm{O}(n^5 \log M)$. □

5. Conclusion.

We have described a combinatorial polynomial algorithm for minimizing integer-valued bisubmodular functions. If we are given a positive lower bound ϵ for the difference between the minimum and the second minimum value of f, a variant of the present algorithm works for any real-valued bisubmodular function f. The only required modification is to change the stopping rule $\delta < 1/3n^2$ to $\delta < \epsilon/3n^2$. The running time is $\mathrm{O}(n^5 \log(M/\epsilon))$. Thus we obtain a polynomial algorithm for testing membership in delta-matroid polyhedra.

One can make this algorithm strongly polynomial with the aid of the generic preprocessing technique of Frank–Tardos [13] that uses the simultaneous Diophantine approximation. However, a more natural strongly polynomial algorithm is desirable. Subsequent to this paper, McCormick and Fujishige [21] have devised such an algorithm for general bisubmodular function minimization.

Acknowledgments. The authors are very grateful to Kazutoshi Ando and an anonymous referee for pointing out errors in earlier versions of this manuscript.

REFERENCES

[1] K. Ando and S. Fujishige, *On structures of bisubmodular polyhedra*, Math. Program., 74 (1996), pp. 293–317.

[2] E. Balas and W. R. Pulleyblank, *The perfectly matchable subgraph polytope of an arbitrary graph*, Combinatorica, 9 (1989), pp. 321–337.

[3] A. Bouchet, *Greedy algorithm and symmetric matroids*, Math. Program., 38 (1987), pp. 147–159.

[4] A. Bouchet, *Matchings and $\triangle$-matroids*, Discrete Appl. Math., 24 (1989), pp. 55–62.

[5] A. Bouchet and W. H. Cunningham, *Delta-matroids, jump systems and bisubmodular polyhedra*, SIAM J. Discrete Math., 8 (1995), pp. 17–32.

[6] R. Chandrasekaran and S. N. Kabadi, *Pseudomatroids*, Discrete Math., 71 (1988), pp. 205–217.

[7] W. H. Cunningham, *Testing membership in matroid polyhedra*, J. Combin. Theory Ser. B, 36 (1984), pp. 161–188.

[8] W. H. Cunningham, *On submodular function minimization*, Combinatorica, 5 (1985), pp. 185–192.

[9] W. H. Cunningham and J. Green-Krótki, *b-matching degree sequence polyhedra*, Combinatorica, 11 (1991), pp. 219–230.

[10] W. H. Cunningham and J. Green-Krótki, *A separation algorithm for the matchable set polytope*, Math. Programming, 65 (1994), pp. 139–150.

[11] A. Dress and T. F. Havel, *Some combinatorial properties of discriminants in metric vector spaces*, Adv. Math., 62 (1986), pp. 285–312.

[12] F. D. J. Dunstan and D. J. A. Welsh, *A greedy algorithm solving a certain class of linear programmes*, Math. Program., 5 (1973), pp. 338–353.

[13] A. Frank and É. Tardos, *An application of simultaneous Diophantine approximation in combinatorial optimization*, Combinatorica, 7 (1987), pp. 49–65.

[14] S. Fujishige, *Submodular Functions and Optimization*, North-Holland, Amsterdam, 1991; 2nd ed., 2005.

[15] S. Fujishige, *A min-max theorem for bisubmodular polyhedra*, SIAM J. Discrete Math., 10 (1997), pp. 294–308.

[16] M. Grötschel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica, 1 (1981), pp. 169–197.

[17] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin, 1988.

[18] S. Iwata, L. Fleischer, and S. Fujishige, *A combinatorial strongly polynomial algorithm for minimizing submodular functions*, J. ACM, 48 (2001), pp. 761-777.

[19] L. Lovász, *Submodular functions and convexity*, in Mathematical Programming — The State of the Art, A. Bachem, M. Grötschel and B. Korte, eds., Springer-Verlag, Berlin, 1983, pp. 235–257.

[20] L. Lovász, *The membership problem in jump systems*, J. Combin. Theory Ser. B, 70 (1997), pp. 45–66.

[21] S. T. McCormick and S. Fujishige, *Better algorithms for bisubmodular function minimization*, manuscript, 2003.

[22] L. Qi, *Directed submodularity, ditroids and directed submodular flows*, Math. Program., 42 (1988), pp. 579–599.

[23] A. Schrijver, *A combinatorial algorithm minimizing submodular functions in strongly polynomial time*, J. Combin. Theory Ser. B, 80 (2000), pp. 346–355.

[24] F. Zhang, *A separation algorithm for b-matching degree-sequence polyhedra*, Math. Oper. Res., 28 (2003), pp. 92–102.