SIAM J. DISCRETE MATH.
Vol. 12, No. 1, pp. 1–5

FLOWS IN UNDIRECTED UNIT CAPACITY NETWORKS*

ANDREW V. GOLDBERG† AND SATISH RAO†

Abstract. We describe an $O(\min(m, n^{3/2})m^{1/2})$-time algorithm for finding maximum flows in undirected networks with unit capacities and no parallel edges. This improves upon the previous bound of Karzanov and Even and Tarjan when $m = \omega(n^{3/2})$, and upon a randomized bound of Karger when $v = \Omega(n^{7/4}/m^{1/2})$.

Key words. algorithms, combinatorial optimization, maximum flow, undirected graphs, sparcification

AMS subject classifications. F.2.2, G.2.2

PII. S089548019733103X

1. Introduction.

In this paper we consider the undirected maximum flow problem in a network with unit capacities and no parallel edges. Until recently, the fastest known way to solve this problem was using a reduction to the directed problem with unit capacities and no parallel arcs. Karzanov [8] and Even and Tarjan [2] have shown that Dinitz's blocking flow algorithm [1], applied to the directed problem, runs in $O(\min(m^{1/2}, n^{2/3})m)$ time. (Here n and m are the number of input vertices and edges, respectively.)

Recently, Karger [6] developed two randomized algorithms for the undirected problem, with running times of $O^*(m^{5/6}n^{1/3}v^{2/3})$ and $O^*(m^{2/3}n^{1/3}v)$. (Here v is the maximum flow value.)

We develop an $O(\min(m, n^{3/2})m^{1/2})$-time algorithm for the problem. This improves the previous deterministic bound for $m = \omega(n^{3/2})$ and Karger's previous randomized bound for $v = \Omega(n^{7/4}/m^{1/2})$. We note that recently, using methods in this paper and improved sampling techniques, Karger [7] developed a randomized algorithm that has $O^*(\sqrt{mnv})$ running time.

Our improvements are based on the sparsification technique of Nagamochi and Ibaraki [9]. Their technique applies to undirected (e.g., symmetric) graphs. We use their technique in the context of residual graphs of flows in undirected graphs, which are not symmetric.

We note that the Nagamochi and Ibaraki sparsification technique has previously been used to improve performance of maximum flow algorithms for undirected unit capacity networks when v is small. In combination with the augmenting path algorithm [3], this technique gives an $O(nv^2)$ time bound. In combination with Karger's second algorithm, this technique gives an $O^*(nv^{5/3})$ expected time bound. See, for example, [4].

2. Preliminaries.

For this paper, we consider computing a maximum flow in an undirected graph $G = (V, E)$ with two distinguished vertices s and t. We consider only zero-one–valued flows. Let $|V| = n$ and $|E| = m$. To define undirected flow, we

*Received by the editors November 21, 1997; accepted for publication (in revised form) April 21, 1998; published electronically January 29, 1999. A preliminary version of this paper appeared in *Proc. 38th IEEE Symposium on Foundations of Computer Science,* Miami Beach, FL, IEEE Computer Society Press, Los Alamitos, CA, 1997, pp. 32–34.

http://www.siam.org/journals/sidma/12-1/33103.html

†NEC Research Institute, 4 Independence Way, Princeton, NJ 08540 (avg@research.nj.nec.com, satish@research.nj.nec.com).

consider the symmetric directed graph defined by G: $G' = (V, E')$ such that $(i,j) \in E'$ if and only if $\{i,j\} \in E$. A *directed flow* in G' is a zero-one function f' on E' obeying *conservation constraints*

$$\sum_{(i,j)\in E'} f'(i,j) = \sum_{(j,k)\in E'} f'(j,k)$$

for all $j \neq s,t$. The *value* of a directed flow f' is defined by $|f'| = \sum_{(i,t)\in E'} f'(i,t) - \sum_{(t,k)\in E'} f'(t,k)$.

A directed flow f' is *proper* if for all $(i,j) \in E'$, $f'(i,j) + f'(j,i) \leq 1$, $f'(i,s) = 0$ for all $(i,s) \in E'$, and $f'(t,k) = 0$ for all $(t,k) \in E'$. A proper flow f' in G' induces an (undirected) *flow* in G in a natural way: $f(\{i,j\}) = f'(i,j) + f'(j,i)$. Without ambiguity, we shall abbreviate $f(\{i,j\})$ to $f(i,j)$. A zero-one function f on E is a flow if f is induced by some proper flow f' in G'. The *value* of f is defined by $|f| = |\{\{i,t\} \in E : f(i,t) = 1\}|$. Our algorithms maintain a flow f in G implicitly by maintaining a proper flow f' in G'.

An s-t cut is a partitioning of vertices $(S, V-S)$ such that $s \in S$ and $t \in V-S$. A *capacity* of an s-t cut $(S, V-S)$ is given by $|\{(i,j) \in E' : i \in S, j \notin S\}|$. An arc $(i,j) \in E'$ is *residual* if $f(i,j) = 0$. A *residual capacity* of an s-t cut in G' under f' is given by $|\{(i,j) \in E' : i \in S, j \notin S, f'(i,j) = 0\}| \cup |\{(j,i) \in E' : i \in S, j \notin S, f(i,j) = 1\}|$. A residual flow value is the difference between the maximum and the current flow values.

Suppose an undirected flow f is induced by a proper flow f'. We denote the length of the shortest residual path from s to t in G' by D_f.

We define the *auxiliary graph* of a proper flow f' as the subset of arcs in G' that lie on any shortest s-t path in G'. Note that this graph is acyclic, since for each edge (u,v) in the graph u must be strictly farther from t than v is.

A directed flow g' in a graph H' is *blocking* if every s-t path in H' contains an arc a with $g'(a) = 1$. In the unit capacity case, a blocking flow in an acyclic graph H' can be found in $O(m)$ time.

Dinitz's algorithm [1] for finding maximum flows in undirected graphs repeatedly augments the current flow by a blocking flow in the graph induced by the residual arcs on shortest paths from s to t. Based on the following two lemmas, Karzanov [8] and Even and Tarjan [2] have shown that Dinitz's algorithm terminates in $\min(n^{2/3}, m^{1/2})$ iterations. Note that these lemmas hold for both directed and undirected flows.

LEMMA 2.1. *Given a network G with no parallel edges and a flow f, the residual flow is at most* $(2n/D_f)^2$ [2, 8].

LEMMA 2.2. *Given a network G and a flow f, the residual flow is at most m/D_f* [2, 8].

3. Network sparsification. We extend the sparsification technique of Nagamochi and Ibaraki [9] to residual graphs as follows. Given a graph $G = (V,E)$ and a flow f, we define E_f^0 to be the subset of edges with zero flow and define $E_f^1 = E - E_f^0$. The following procedure removes some edges in E_f^0 from E.

SPARSIFY(v, G, f)

1. Find $E_1, E_2, \ldots, E_v$, where E_i is a maximal spanning forest in $(V, E_f^0 - \cup_{j<i} E_j)$.
2. Output $(V, E_f^1 + \cup_{j \leq v} E_j)$.

The following lemma follows from [9].

LEMMA 3.1. *There is an $O(m)$ implementation of* SPARSIFY.

SPARSIFY is defined so that the following lemma holds. The proof of the lemma is analogous to the proof of Lemma 2.1 in [10].

LEMMA 3.2. *The residual flow in the network output by* SPARSIFY(v, G, f) *is at least* $\min(v, r)$*, where* r *is the residual flow value in* G_f*.*

Proof. Let $G_i = (V, E_f - E_f^0 + \cup_{j \le i} E_i)$. We prove that the maximum s-t flow in G_i is at least $\min(i, r)$.

The claim is clearly true for G_0. We assume it to be true for G_i: the residual flow in G_i with respect to f is at least $\min(i, r)$. Assume that the residual flow in G_{i+1} with respect to f is less than $\min(i + 1, r)$. Since G_i is contained in G_{i+1}, this means that the residual flow with respect to f is equal to i and that $r \ge i + 1$.

By the maximum flow–minimum cut theorem, there is an s-t cut, $(S, \overline{S})$, with residual capacity i in G_{i+1} with respect to f. Moreover, the residual capacity of the cut in G_i with respect to f is also i. Thus, there is no edge in E_{i+1} that crosses the cut.

On the other hand, the capacity of the cut is at least $i+1$ in G_f. Thus, there is an undirected edge that is not in G_i that is in G_f that crosses the cut. This contradicts the assumption that E_{i+1} is a maximal spanning forest of $(V, E_f^0 - \cup_{j<i+1} E_j)$.

The residual flow in G_{i+1} with respect to f must therefore be at least min $(i + 1, r)$. ∎

4. The algorithm. Now we are ready to describe our main algorithm. The algorithm works with sparser and sparser graphs. We denote the current graph by $\overline{G} = (V, \overline{E})$ and $|\overline{E}|$ by $\overline{m}$.

The algorithm is based on the SPARSIFYANDBLOCK step. This step applies the procedure SPARSIFY$((2n/D_f)^2, \overline{G}, f)$ to obtain a new current graph $\overline{G}$ and augments f by a blocking flow in the auxiliary graph of $\overline{G}_f$. The SPARSIFYANDBLOCK step takes $O(m)$ time. Initially, $\overline{G} = G$ and f is the zero flow. The algorithm repeatedly applies the SPARSIFYANDBLOCK step until f is a maximum flow in $\overline{G}$.

First we show that the algorithm is correct.

LEMMA 4.1. *The algorithm terminates with a maximum flow in* G*.*

Proof. The algorithm terminates because each iteration increases the value of f, and the fact that the algorithm finds a maximum flow follows from Lemmas 2.1 and 3.2. ∎

Next we analyze the algorithm. Note that the algorithm's bounds are at least as good as those for the blocking flow algorithm because the work of SPARSIFY is linear per blocking flow computation. Because of the sparsification, our algorithm works with smaller graphs and may be faster.

Remark. During initial iterations when D_f is small and $n(2n/D_f)^2 \ge m$, SPARSIFY does not delete any edges and may be omitted. Toward the end, when D_f is large and $|E_f^1| \ge n(2n/D_f)^2$, SPARSIFY does not significantly reduce the number of edges and may be omitted as well.

LEMMA 4.2. *At any point during the algorithm,* $|E_f^1| \le 8n^{3/2}$.

Proof. The size of E_f^1 is upper-bounded by the total length of augmenting paths in a blocking flow algorithm. We proceed by showing, as in [2], that this bound is as stated in the lemma.

The algorithm performs at most $(n/d)^2$ augmentations when $2d \le D_f < 4d$. Note that the algorithm always augments flow on shortest paths. Thus, the total length of the augmenting paths after the length of the augmentations is at least $2d_0$ and is bounded by $\sum_{i \ge 0} (4n^2/(2^i d_0))$.

Since the total s-t flow in a graph with no parallel edges is at most n, the total length of the augmenting paths is at most $2nd_0 + \sum_{i\geq 0}(4n^2/(2^i d_0)) = 2nd_0 + 8n^2/d_0$. Choosing $d_0 = 2n^{1/2}$ proves the lemma. ∎

Notice that this lemma implies that $\overline{m} \leq 8n^{3/2} + n(2n/D_f)^2$ throughout the algorithm.

THEOREM 4.3. *The algorithm runs in time* $O(\min(m, n^{3/2})m^{1/2})$.

Proof. If $m = O(n^{3/2})$, then by [2, 8], the running time is $O(m^{3/2})$, which is $O(n^{3/2}m^{1/2})$ for these values of m. For the rest of the proof we assume $m = \Omega(n^{3/2})$.

During the initial iterations of the algorithm, when $n(2n/D_f)^2 > m$, the sparsification has no effect and each iteration takes $O(m)$ time. During these iterations, $D_f < 2n^{3/2}/m^{1/2}$. Since D_f increases at each iteration, these iterations take a total of $O(n^{3/2}m^{1/2})$ time.

During the final iterations of the algorithm, when $n(2n/D_f)^2 \leq 4n^{3/2}$, we have $\overline{m} = O(n^{3/2})$ by Lemma 4.2 and, by [2, 8], these iterations take $O(n^{9/4})$ time. This is $O(n^{3/2}m^{1/2})$ for $m = \Omega(n^{3/2})$.

Finally, we account for the remaining iterations where $\sqrt{2}n^{3/2}/m^{1/2} < D_f \leq n^{3/4}$. Each iteration takes $O(\overline{m})$ time. If during an iteration $D_f = i$, then

$$\overline{m} \leq n(2n/i)^2 + 8n^{3/2} \leq 2n\left(\frac{2n}{i}\right)^2.$$

Let $D_0 = 2n^{3/2}/m^{1/2}$. The total work during these iterations is at most

$$\sum_{i=D_0}^{\infty} 2n(2n/i)^2 = 8n^3 \sum_{i=D_0}^{\infty} \frac{1}{i^2}.$$

We bound the sum as follows:

$$\sum_{i=D_0}^{\infty} \frac{1}{i^2} \leq \frac{1}{D_0^2} \sum_{i=D_0}^{\infty} \frac{1}{\lfloor i/D_0 \rfloor^2} \leq \frac{1}{D_0} \sum_{i=1}^{\infty} \frac{1}{i^2} \leq \frac{1}{D_0}\frac{\pi^2}{6}.$$

Therefore, the work is $O(n^{3/2}m^{1/2})$. ∎

5. Concluding remarks. We have shown how to use the sparsification technique of Nagamochi and Ibaraki to speed up the blocking flow method on dense networks with unit capacities and no parallel edges. Because the sparsification is quite efficient, our approach may be practical, especially if the push-relabel algorithm of [5] is used instead of the blocking flow method. It would be interesting to verify the practicality experimentally.

Acknowledgment. We would like to thank Mikkel Thorup for comments on a draft of this paper.

REFERENCES

[1] E. A. DINIC, *Algorithm for solution of a problem of maximum flow in networks with power estimation*, Soviet Math. Dokl., 11 (1970), pp. 1277–1280.

[2] S. EVEN AND R. E. TARJAN, *Network flow and testing graph connectivity,* SIAM J. Comput., 4 (1975), pp. 507–518.

[3] L. R. FORD, JR. AND D. R. FULKERSON, *Flows in Networks,* Princeton University Press, Princeton, NJ, 1962.

[4] H. N. Gabow, *A matroid approach to finding edge connectivity and packing arborescences,* J. Comput. System Sci., 50 (1995), pp. 259–273.

[5] A. V. Goldberg and R. Kennedy, *Global price updates help,* SIAM J. Discrete Math., 10 (1997), pp. 551–572.

[6] D. R. Karger, *Using random sampling to find maximum flows in uncapacitated undirected graphs,* in Proc. 29th Annual ACM Symposium on Theory of Computing, ACM, New York, 1997, pp. 240–249.

[7] D. R. Karger, *Better random sampling algorithms for flows in undirected graphs,* in Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, Howard Karloff, ed., SIAM, Philadelphia, PA, 1998, pp. 490–499.

[8] A. V. Karzanov, *On finding maximum flows in networks with special structure and some applications,* in Matematicheskie Voprosy Upravleniya Proizvodstvom 5, Moscow State University Press, Moscow, 1973, pp. 66–70 (in Russian).

[9] H. Nagamochi and T. Ibaraki, *A linear-time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph*, Algorithmica, 7 (1992), pp. 583–596.

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

SIAM J. DISCRETE MATH.
Vol. 12, No. 1, pp. 6–26

A LINEAR TIME ALGORITHM FOR EMBEDDING GRAPHS IN AN ARBITRARY SURFACE*

BOJAN MOHAR†

Abstract. For an arbitrary fixed surface S, a linear time algorithm is presented that for a given graph G either finds an embedding of G in S or identifies a subgraph of G that is homeomorphic to a minimal forbidden subgraph for embeddability in S. A side result of the proof of the algorithm is that minimal forbidden subgraphs for embeddability in S cannot be arbitrarily large. This yields a constructive proof of the result of Robertson and Seymour that for each closed surface there are only finitely many minimal forbidden subgraphs. The results and methods of this paper can be used to solve more general embedding extension problems.

Key words. surface embedding, obstruction, algorithm, graph embedding, forbidden subgraph, forbidden minor

AMS subject classifications. 05C10, 05C85, 68Q20, 68R10

PII. S089548019529248X

1. Introduction. The problem of constructing embeddings of graphs in surfaces is of practical and theoretical interest. The practical issues arise, for example, in problems concerning VLSI and also in several other applications since graphs embedded in low genus surfaces can be handled more easily. Theoretical interest comes from the importance of the genus parameter of graphs and from the fact that graphs of bounded genus naturally generalize the family of planar graphs and share many important properties with them.

There are linear time algorithms that for a given graph determine whether the graph can be embedded in the 2-sphere (or in the plane). The first such algorithm was obtained by Hopcroft and Tarjan [16] in 1974. There are several other linear time planarity algorithms (Booth and Lueker [6], de Fraysseix and Rosenstiehl [11], Williamson [36, 37]). Extensions of these algorithms return an embedding (rotation system) whenever a graph is found to be planar [7] or exhibit a forbidden Kuratowski subgraph homeomorphic to K_5 or $K_{3,3}$ if the graph is nonplanar [36, 37] (see also [21]). Recently, linear time algorithms have been devised for embedding graphs in the projective plane (Mohar [22]) and in the torus (Juvan, Marinček, and Mohar [19]).

It is known that the general problem of determining the genus [34] or the nonorientable genus [35] of graphs is NP-hard. However, for every fixed surface there is a polynomial time algorithm which checks if a given graph can be embedded in the surface. Such algorithms were found first by Filotti, Miller, and Reif [10]. For a fixed orientable surface S of genus g they discovered an algorithm with time complexity $O(n^{\alpha g+\beta})$ (α, β are constants), which tests if a given graph of order n can be embedded in S. Unfortunately, the degree of the polynomial time complexity is very large, even in the simplest case when S is the torus. A theoretical estimate on the running time in case of the torus is only $O(n^{188})$. Recently, Djidjev and Reif [9] announced improvement of the algorithm in [10] by presenting a polynomial time algorithm for

*Received by the editors September 27, 1995; accepted for publication (in revised form) June 4, 1998; published electronically January 29, 1999. This research was supported in part by the Ministry of Science and Technology of Slovenia, Research Project P1–0210–101–94.
http://www.siam.org/journals/sidma/12-1/29248.html

†Department of Mathematics, University of Ljubljana, Jadranska 19, 1111 Ljubljana, Slovenia (bojan.mohar@uni-lj.si).

each fixed orientable surface, where the degree of the polynomial is fixed. The basic technique used in [10] and in [9] of embedding a subgraph, attempting to extend this partial embedding, and recursively working with discovered forbidden subgraphs for smaller genus surfaces is also used in our algorithm.

For every fixed surface S, an $O(n^3)$ algorithm for testing embeddability in S can be devised using graph minors [27, 31]. Robertson and Seymour recently improved their $O(n^3)$ algorithms to $O(n^2 \log n)$ [28, 29, 30]. An extension which also constructs an embedding is described by Archdeacon in [2]. The running time is estimated to be $O(n^{10})$ but with a little additional care it could be decreased to $O(n^6)$. A disadvantage of these algorithms is that they use the lists of forbidden minors which are not known for surfaces different from the 2-sphere and the projective plane. Even for the projective plane whose forbidden minors are known [1, 13], the algorithms based on checking for the presence of forbidden minors are rather time consuming since their running time estimates involve enormous constants.

In this paper we describe a linear time algorithm which finds an embedding of a given graph G into a surface S if such an embedding exists. Here S is an arbitrary fixed surface. In the case when G cannot be embedded in S, the algorithm returns a subgraph H of G that cannot be embedded in S, but every proper subgraph of H admits an embedding in S. A side result of the algorithm is that the returned "minimal forbidden subgraph" H is homeomorphic to a graph with a bounded number of edges (where the bound depends only on S). This yields a constructive proof of the result of Robertson and Seymour [27] that for each closed surface there are only finitely many minimal forbidden subgraphs. A constructive proof for nonorientable surfaces has been published by Archdeacon and Huneke [3], while orientable surfaces resisted all previous attempts. (Recently Seymour [32] also found a constructive proof of that result.)

The results and methods of this paper can be used toward solving a generalization of problems of embedding graphs in surfaces—the so-called *embedding extension problems* where one has a fixed embedding of a subgraph K of G in some surface and asks for embedding extensions to G or (minimal) obstructions for existence of such extensions.

The paper is more or less self-contained with the exception of using results from [17, 18, 20, 24].

Concerning the time complexity of our algorithms, we assume a random-access machine (RAM) model with unit cost for some basic operations. This model of computation was introduced by Cook and Reckhow [8]. It is known as the *unit-cost* RAM where operations on integers, whose value is $O(n)$, need only constant time (n is the order of the given graph). The same model of computation is used in many other instances, for example, in well-known linear time planarity testing algorithms [16].

2. Basic definitions. We follow standard graph theory terminology as used, for example, in [5]. Let G and H be graphs. We denote by $G - H$ the graph obtained from G by deleting all vertices of $G \cap H$ and all their incident edges. If $F \subseteq E(G)$, then $G - F$ denotes the graph obtained from G by deleting all edges in F.

We will consider 2-cell embeddings of graphs in closed surfaces. They can be described in a purely combinatorial way by specifying the following:

(1) A *rotation system* $\pi = (\pi_v \,;\, v \in V(G))$. For each vertex v of the given graph G we have a cyclic permutation π_v of edges incident with v, representing their circular order around v on the surface.
(2) A *signature* $\lambda : E(G) \to \{-1, 1\}$. Suppose that $e = uv$. Following the edge e on the surface, we see if the local rotations π_v and π_u are chosen consistently

or not. If yes, then we have $\lambda(e) = 1$; otherwise we have $\lambda(e) = -1$.

The reader is referred to [14] or [25] for more details. We will use this description as a definition: An *embedding* of a connected graph G is a pair $\Pi = (\pi, \lambda)$, where π is a rotation system and λ is a signature. Having an embedding Π of G, we say that G is Π-*embedded*. If H is a subgraph of G, then the *induced embedding* of H (or the *restriction* of Π to H) is obtained from that of G by ignoring all edges in $E(G)\backslash E(H)$ and by restricting the signature to $E(H)$.

Each embedding Π of G determines a set of closed walks in G, called Π-*facial walks* or simply Π-*faces*, that correspond to traversals of face boundaries of the corresponding topological embedding. Either each edge e of G is contained in exactly two Π-facial walks, or it appears twice in the same Π-facial walk W. In the latter case, e and W are said to be *singular*. Edges e and f incident with the same vertex v of G are Π-*consecutive* if $e = \pi_v(f)$ or $f = \pi_v(e)$. In that case, there is a Π-face F containing e and f as consecutive edges, and we say that the pair $\{e, f\}$ is an *angle* of F.

Suppose that a subgraph K of G is Π-embedded. An embedding $\tilde{\Pi}$ of G is an *extension* of Π if it is an embedding in the same surface as Π and the induced embedding of K is equal to Π. Given a graph G and a Π-embedded subgraph K, we may ask if there is an embedding extension to G. This problem will be referred to as an *embedding extension problem*. An *obstruction* for extensions for such a problem is a subgraph Ω of $G - E(K)$ such that no embedding extension of K to $K \cup \Omega$ exists.

3. Bridges. Let K be a subgraph of G. A K-*bridge* in G (or a *bridge* of K in G) is a subgraph of G which is either an edge $e \in E(G)\backslash E(K)$ with both endpoints in K or a connected component of $G - V(K)$ together with all edges (and their endpoints) between this component and K. Each edge of a K-bridge B having an endpoint in K is a *foot* of B. The vertices of $B \cap K$ are the *vertices of attachment* of B, and B is *attached to* each of these vertices. A vertex of K of degree different from 2 is a *main vertex* (or a *branch vertex*) of K. For convenience, if a connected component C of K is a cycle, then we choose an arbitrary vertex of C and declare it to be a main vertex of K as well. A *branch* of K is any path in K (possibly closed) whose endpoints are main vertices but no internal vertex on this path is a main vertex. Every subpath of a branch e is a *segment* of e. If a K-bridge is attached to a single branch e of K, it is said to be *local* (on e). The number of branches of K, denoted by $\mathrm{bsize}(K)$, is the *branch size* of K. If B is a K-bridge in G, then the *size* $\mathrm{bsize}_K(B)$ of B is defined as the number of branches of $K \cup B$ that are contained in B. Note that $\mathrm{bsize}(K \cup B) \le \mathrm{bsize}(K) + 2\,\mathrm{bsize}_K(B)$. A *basic piece* of K is either a main vertex or an *open branch* of K (i.e., a branch with its endpoints removed). If a K-bridge B in G is attached to at least three basic pieces of K, then B is *strongly attached*. Otherwise, it is *weakly attached*.

Suppose that K is Π-embedded. Let B be a K-bridge in G and $\tilde{\Pi}$ an extension of Π to $K \cup B$. Then there is a unique Π-face F that is not a $\tilde{\Pi}$-face, and we say that B is *embedded in* F or that F *contains* B. Clearly, if B is embedded in F, then all basic pieces that B is attached to appear on F. Each basic piece on F has one or more *appearances* (or *occurrences*) on F. The total number of appearances of main vertices on F is the *branch size* of F. We say that the K-bridge B embedded in F is *attached to* an appearance of the basic piece x on F if x contains a vertex x_0 such that the angle in F at this appearance of x_0 on x is not an angle within a $\tilde{\Pi}$-face.

LEMMA 3.1. *Suppose that there are no local K-bridges in G. Let $\tilde{\Pi}$ be an embedding of G that is an extension of an embedding Π of K. If B is a K-bridge embedded* *in a* Π*-face* F*, we denote by* $q(B)$ *the number of appearances of basic pieces on* F *that* B *is attached to. If* F *is a* Π*-face of branch size* s*, and* $B_1, \ldots, B_k$ *are* K*-bridges embedded in* F*, then*

$$\sum_{i=1}^{k}(q(B_i) - 2) \le 2s - 2\,. \tag{3.1}$$

Consequently, if $\mathcal{B}$ *is the set of all* K*-bridges in* G*, then*

$$\sum_{B\in\mathcal{B}}(q(B) - 2) \le 4\,\mathrm{bsize}(K). \tag{3.2}$$

Proof. The proof of (3.1) is by induction on the number $p \le 2s$ of those occurrences of basic pieces on F that some bridge is attached to. We can assume that $q(B_i) \ge 3$ for $1 \le i \le k$ and that $p \ge 2$. The case $p = 2$ is trivial. If $p > 2$, let B be a strongly attached bridge in F. Let $f_1, \ldots, f_q$ be feet of B attached to distinct basic pieces of K. They divide F into q segments, containing $p_1, \ldots, p_q$ appearances of basic pieces of K (or their parts), respectively. Clearly, $p_1 + \cdots + p_q = p + q$ and $p_i < p$, $i = 1, \ldots, q$. By the induction hypothesis

$$\sum_{i=1}^{k}(q(B_i) - 2) \le (p_1 - 2) + \cdots + (p_q - 2) + (q - 2) = p - 2.$$

This proves (3.1). The sum of the branch sizes of Π-faces equals $2\,\mathrm{bsize}(K)$. Hence, (3.2) follows from (3.1). □

Lemma 3.1 shows, in particular, that too many strongly attached bridges obstruct embedding extensions. Similarly, every weakly attached bridge that is embedded such that it is attached to two or more occurrences of the same basic piece contributes to the left side of (3.2). Thus, under an embedding extension all except a bounded number of bridges are attached to at most one appearance of the same basic piece. Such embeddings of bridges are *simple*. More generally, and embedding extension is *simple* if all bridges have simple embeddings. In case of simple embeddings, we will use some special subgraphs of K-bridges in G. If B is a K-bridge in G, an E-*graph* in B is a minimal subgraph H of B such that we have the following:

(E1) Any two vertices of $H - K$ are connected by a path in $H - K$.

(E2) For each branch vertex ζ that B is attached to, H contains a foot incident with ζ. If ζ is an open branch with ends x_1 and x_2 and B is attached to ζ, let ζ_i be the vertex of attachment of B on ζ which is closest to x_i $(i = 1, 2)$. Then H contains feet attached to ζ_1 and ζ_2, respectively (possibly just one if $\zeta_1 = \zeta_2$).

(E3) Every simple extension of any embedding of K to $K \cup H$ determines a simple extension to $K \cup B$.

The difficult part of this paper is to discover obstructions for simple embedding extensions. The next result somehow simplifies this problem by showing that one can work only with E-graphs of K-bridges in G and that E-graphs are not too large.

THEOREM 3.2 (Mohar [24]). *Let* $\mathcal{B}$ *be the set of* K*-bridges in* G*. There is a number* c *depending only on* $\mathrm{bsize}(K)$ *such that each* $B \in \mathcal{B}$ *contains an E-graph* $\tilde{B}$ *with* $\mathrm{bsize}_K(\tilde{B}) \le \mathrm{c}$*. If* $\{B_1, \ldots, B_k\} \subseteq \mathcal{B}$ $(k \ge 1)$ *are arbitrary nonlocal* K*-bridges,* $\tilde{B}_1, \ldots, \tilde{B}_k$ *their corresponding E-graphs, and if* Π *is an embedding of* K*, then any simple extension of* Π *to* $K \cup \tilde{B}_1 \cup \cdots \cup \tilde{B}_k$ *can be further extended to a simple extension* *of* Π *to* $K \cup B_1 \cup \cdots \cup B_k$. *Moreover, there is a linear time algorithm that replaces all* K*-bridges* B *in* G *with their E-graphs* $\tilde{B}$.

In [24] it is further proved that the size of E-graphs of weakly attached bridges is at most 12. Moreover, if a weakly attached bridge B has some simple embedding extension, then $\mathrm{bsize}_K(\tilde{B}) \le 5$.

Theorem 3.2 shows that we can replace every K-bridge B in G by its small E-graph $\tilde{B}$, and simple embedding extension problems do not change. This enables us to consider only obstructions that can be expressed as the union of E-graphs.

4. Restricted embedding extensions. Let K be a subgraph of G and let $\mathcal{P}$ be the set of all basic pieces of K. If B is a K-bridge, let $T \subseteq \mathcal{P}$ be the set of basic pieces of K that B is attached to. We say that B is of *type* T. Suppose that K is Π-embedded in some surface. In general, a bridge of type T can be embedded in two or more faces of K, and in some faces in several different ways. To formalize the essentially different ways of embedding bridges in particular faces, we introduce the notion of embedding schemes. Let F be a Π-face. For $T \subseteq \mathcal{P}$, let $\pi_1, \ldots, \pi_k$ be the appearances of basic pieces from T on F. An *embedding scheme* for the type T in the face F is a subset of $\pi_1, \ldots, \pi_k$ in which at least one appearance of every basic piece from T occurs. An embedding scheme δ is *simple* if each basic piece from T has exactly one appearance in δ. There is a natural partial ordering among the embedding schemes for the type $T \subseteq \mathcal{P}$ in F, induced by the set inclusion: If δ and δ' are embedding schemes for T in the same face F, then $\delta \preceq \delta'$ if every appearance of a basic piece in δ also participates in δ'.

Let B be a K-bridge of type T and δ an embedding scheme for T in a face F. An embedding of B in F is δ*-compatible* (for short, a δ*-embedding*) if B is attached only to appearances of basic pieces from δ. If $\delta \preceq \delta'$, then every δ-embedding is also a δ'-embedding.

An *embedding distribution* $\Delta(T)$ *for a type* $T \subseteq \mathcal{P}$ is a selection of embedding schemes for the type T, possibly in different faces. Suppose that $T_1, T_2, \ldots, T_s$ are all types of K-bridges in G. An *embedding distribution* is a family $\Delta = \{\Delta(T_1), \ldots, \Delta(T_s)\}$, where $\Delta(T_i)$ is an embedding distribution for the type T_i, $i = 1, \ldots, s$. Δ is *simple* if all embedding schemes in $\Delta(T_1), \ldots, \Delta(T_s)$ are simple. Let $\mathcal{B}$ be a set of K-bridges with an embedding extending the given embedding of K. We say that the embedding of $\mathcal{B}$ is Δ*-compatible* (or a Δ*-embedding*) if the embedding of each bridge $B \in \mathcal{B}$ is δ-compatible for some $\delta \in \Delta(T)$, where T is the type of B. The relation $\preceq$ naturally extends from embedding schemes to embedding distributions. The *order* $\mathrm{ord}(\Delta)$ of Δ is equal to the total number of embedding schemes in the embedding distributions $\Delta(T_i)$, $i = 1, \ldots, s$. If Δ is an embedding distribution for the set $\mathcal{B}$ of K-bridges in G and if $\mathcal{B}' \subseteq \mathcal{B}$, then the *restriction* of Δ to $\mathcal{B}'$ is the embedding distribution obtained from Δ by removing the embedding distributions $\Delta(T)$ for those types T that are not present among the bridges in $\mathcal{B}'$. If there is no confusion, the restriction of Δ to $\mathcal{B}'$ is also denoted by Δ.

Now we introduce a formal definition of an *embedding extension problem*, abbreviated EEP. This is a quadruple $\Xi = (G, K, \Pi, \Delta)$ where G is a graph, K is a subgraph of G, Π is an embedding of K, and Δ is an embedding distribution for the K-bridges in G. The EEP is *simple* if Δ is simple. An *embedding extension* (EE) for Ξ is an embedding extension of Π to G such that every K-bridge is Δ-embedded. An *obstruction* for Ξ is a set $\mathcal{B}$ of K-bridges or their subgraphs such that $(K \cup \mathcal{B}, K, \Pi, \Delta)$

admits no EE. The *size* $\mathrm{bsize}_K(\mathcal{B})$ of an obstruction $\mathcal{B}$ is

$$\mathrm{bsize}_K(\mathcal{B}) = \sum_{B\in\mathcal{B}} \mathrm{bsize}_K(B).$$

Embedding distributions will be used in later discussion in the following way. For every possible embedding distribution Δ we will try to extend the given embedding of K to a Δ-embedding of G. Embedding distributions will be selected one after another respecting the order $\preceq$. We start with the embedding distribution of order 0, and any bridge is an obstruction for this subproblem. In a general step, we already have obstructions for all embedding distributions $\Delta' \prec \Delta$. Let $\mathcal{B}$ denote their union. Then we try to extend each Δ-embedding of $\mathcal{B}$ to a Δ-embedding of G. Obtaining an embedding, we stop and return the embedding (and our task is complete). Otherwise, an obstruction is obtained. Finally, the obstructions for different embeddings of $\mathcal{B}$ are combined together with $\mathcal{B}$ into a single obstruction for Δ-compatible embedding extensions. We will refer to this process as the *procedure of embedding distribution of types*.

Suppose that we fix an embedding distribution Δ_0. Using the procedure of embedding distribution of types we determine all (minimal) embedding distributions $\Delta \preceq \Delta_0$ for which a Δ-compatible EE exists, and at the same time construct obstructions for all other Δ-embeddings ($\Delta \preceq \Delta_0$). Algorithmically, a problem in the procedure of embedding distribution of types is in bounding the number of Δ-compatible embeddings of the union $\mathcal{B}$ of obstructions for all simpler embedding distributions. By using an operation called *compression* (cf. section 5), we will be able to prove that all obstructions have bounded size and hence also a bounded number of embeddings. We shall use this approach in the proof of Corollary 5.5.

The procedure of embedding distribution of types can be generalized by introducing the union of EEPs. Suppose that we want to consider embedding extensions where we fix embeddings of some of the bridges. To formalize, we call an EEP $\Xi' = (G, K', \Pi', \Delta')$ a *subproblem* of $\Xi = (G, K, \Pi, \Delta)$ if

(i) K' is the union of K and a set $\mathcal{B}$ of K-bridges in G.
(ii) Π' is an EE of Π.
(iii) the Π'-embedding of every $B \in \mathcal{B}$, viewed as an extension of Π, is Δ-compatible.
(iv) every Δ'-compatible embedding of a K'-bridge in G, viewed as an EE of the embedding Π, is Δ-compatible.

For $i = 1, \ldots, N$, let $\Xi_i = (G, K_i, \Pi_i, \Delta_i)$ be subproblems of $\Xi = (G, K, \Pi, \Delta)$. Denote by $\mathcal{B}_i$ the set of K-bridges in K_i. We say that Ξ is the *union* of subproblems Ξ_i ($1 \le i \le N$) if for every set $\mathcal{B} \supseteq \cup_{i=1}^N \mathcal{B}_i$ of K-bridges in G, the restriction of Ξ to $K \cup \mathcal{B}$ admits an EE exactly when the restriction to $K \cup \mathcal{B}$ of at least one of Ξ_i does. In this case, an EE for some Ξ_i is also an EE for Ξ, while having obstructions Ω_i for Ξ_i ($1 \le i \le N$), their combination

$$\Omega = \bigcup_{i=1}^{N} (\Omega_i \cup \mathcal{B}_i) \tag{4.1}$$

is an obstruction for Ξ.

A subproblem $\Xi' = (G, K, \Pi, \Delta')$ of $\Xi = (G, K, \Pi, \Delta)$ is *equivalent* to Ξ if for every set $\mathcal{B}$ of K-bridges in G and every Δ-compatible EE of K to $K \cup \mathcal{B}$, there is also a Δ'-compatible EE of K to $K \cup \mathcal{B}$. In such a case, an EE for Ξ' is also an EE for Ξ, and every obstruction for Ξ' is an obstruction for Ξ. Therefore, a solution for Ξ' also provides a solution for Ξ.

We shall use the introduced notions mainly in the following particular case.

LEMMA 4.1. *Let* $\Xi = (G, K, \Pi, \Delta)$ *be an EEP. Let* $\mathcal{B}$ *be a set of* K*-bridges in* G*, and let* $\Pi_1, \ldots, \Pi_N$ *be all* Δ*-embeddings of* $\mathcal{B}$ *extending* Π. *For* $i = 1, \ldots, N$, *let* Δ_i *be the largest embedding distribution for* $(K \cup \mathcal{B})$*-bridges in* G *such that every* Δ_i*-embedding of a* $(K \cup \mathcal{B})$*-bridge is also a* Δ*-embedding, and let* $\Delta_i' \preceq \Delta_i$ *be such an embedding distribution that the EEP* $\Xi_i = (G, K \cup \mathcal{B}, \Pi_i, \Delta_i')$ *is equivalent to* $(G, K \cup \mathcal{B}, \Pi_i, \Delta_i)$. *Then* Ξ *is the union of subproblems* $\Xi_1, \ldots, \Xi_N$. *In particular, by solving EEPs* $\Xi_1, \ldots, \Xi_N$ *either we get an EE for* Ξ, *or* (4.1) *gives an obstruction.*

In our algorithms we shall use Lemma 4.1 only in cases when the number of bridges in $\mathcal{B}$ (and hence also the number N of their Δ-embeddings) is bounded by some constant.

We shall also need the following strengthening of a particular case of Lemma 4.1. Let $\Xi = (G, K, \Pi, \Delta)$ be an EEP and x, y be basic pieces (or segments of basic pieces) of K. Denote by $\mathcal{B}_{x,y}$ the set of K-bridges in G of type $T = \{x, y\}$, and suppose that $\mathcal{B}_{x,y} \neq \emptyset$. If x is a main vertex, put $x_1 = x_2 = x$. If x is an open branch, let x_1 and x_2 be vertices of attachment of bridges in $\mathcal{B}_{x,y}$ that are as close as possible to one and the other end of x, respectively. Define similarly y_1 and y_2. For $i, j \in \{1, 2\}$, we select a bridge $B_{x,y}^{i,j} \in \mathcal{B}_{x,y}$ with the following properties:

(a) $B_{x,y}^{i,j}$ is attached to x_i.

(b) Among all bridges from $\mathcal{B}_{x,y}$ attached to x_i, $B_{x,y}^{i,j}$ has an attachment on y as close to y_j as possible.

(c) Subject to (a) and (b), we select $B_{x,y}^{i,j}$ to be an edge if possible.

Let $\mathcal{B}_{x,y}^{\circ}$ be the set of bridges that contains all bridges $B_{x,y}^{i,j}$ $(i, j \in \{1, 2\})$ and for each $\delta \in \Delta(T)$ such that $\mathcal{B}_{x,y}$ has no δ-embedding, $\mathcal{B}_{x,y}^{\circ}$ contains a pair of bridges from $\mathcal{B}_{x,y}$ whose δ-embeddings overlap. If $\Delta(T)$ is simple, then one can construct $\mathcal{B}_{x,y}^{\circ}$ in linear time by using [23].

LEMMA 4.2. *Assuming the above notation, suppose that* $\Delta(T) = \{\delta_1, \delta_2\}$. *Then* Ξ *is equivalent to the union of subproblems* $\Xi' = (G, K \cup \mathcal{B}_{x,y}^{\circ}, \Pi', \Delta')$, *taken over all* Δ*-compatible EEs* Π' *of* Π *to* $K \cup \mathcal{B}_{x,y}^{\circ}$, *where* Δ' *is the restriction of* Δ *to the remaining bridges with the only exception that* $\Delta'(T)$ *contains only those embedding scheme(s)* δ_i $(i \in \{1, 2\})$ *which are used by the bridges from* $\mathcal{B}_{x,y}^{\circ}$ *under the EE* Π'.

Proof. It is to be observed only that whenever the embedding of $\mathcal{B}_{x,y}^{\circ}$ uses just one embedding scheme—say, δ_1—then all bridges from $\mathcal{B}_{x,y}$ may be assumed to be δ_1-embedded since their embedding obstructs possible embeddings of other bridges no more than the embedding of $\mathcal{B}_{x,y}^{\circ}$. □

Let $\Xi = (G, K, \Pi, \Delta)$ be an EEP. Let $\mathcal{B}$ be the set of all K-bridges in G. Suppose that $\mathcal{B} = \mathcal{B}_1 \cup \cdots \cup \mathcal{B}_N$. Denote by Δ_i the restriction of Δ to $\mathcal{B}_i$, $i = 1, \ldots, N$. The EEP $\Xi_i = (K \cup \mathcal{B}_i, K, \Pi, \Delta_i)$ is a *partial problem* of Ξ. We say that Ξ is the *intersection* of partial problems Ξ_i, $i = 1, \ldots, N$, if arbitrary EEs for $\Xi_1, \ldots, \Xi_N$ determine an EE Π_0 for Ξ. More precisely, if there are EEs Π_i for Ξ_i $(i = 1, \ldots, N)$, then there is an EE Π_0 for Ξ such that its restriction to $K \cup \mathcal{B}_i$ coincides with Π_i, $i = 1, \ldots, N$.

Having $\Pi_1, \ldots, \Pi_N$, one can determine Π_0 in linear time as described below. We shall assume that $\operatorname{bsize}(K)$ and N are bounded by a constant since this will hold in our applications (although this assumption is not essential). The number of Π-faces is bounded by $2 \operatorname{bsize}(K)$. Therefore, it suffices to describe the algorithm for an arbitrary Π-face F of K. Let $\mathcal{B}_i' \subseteq \mathcal{B}_i$ $(1 \leq i \leq N)$ be the bridges that are Π_i-embedded in F. Select an orientation of F. For $B \in \mathcal{B}_i'$, let $v_0, \ldots, v_{q-1}$ be its consecutive attachments on F. If e is a foot of B attached to v_j, then we put

$\mathrm{next}(e) = v_{j+1}$ where the index is taken modulo q. The function next can easily be computed in linear time (for all bridges at the same time). Now, consider an appearance of a vertex v on F, and let $\{e_1, e_2\}$ be the angle on F at this appearance of v. We may also assume that F is oriented so that e_1 precedes e_2. The local rotation Π_0 at v between e_1 and e_2 is now easily determined by a merging: we proceed through the lists $L_i = (\Pi_i(e_1), \Pi_i^2(e_1), \Pi_i^3(e_1), \ldots, e_2)$, $i = 1, \ldots, N$, and insert in the rotation of Π_0 at v the initial edge e from that list L_i which has the largest $\mathrm{next}(e)$, i.e., the distance along F from v to $\mathrm{next}(e)$ (in the given direction) is maximal. If there is more than one candidate for e, there are exactly two of them, and one of them belongs to a K-bridge with more than two attachments, the other to a bridge with two attachments. In such a case we select the former one. It can be shown that this procedure gives the desired embedding Π_0. The details are left to the reader.

5. Simple embedding extensions. In this section we will consider only simple embeddings of bridges and simple EEPs. We may assume the following:

(a) Each bridge has been replaced by its small E-graph (cf. Theorem 3.2).
(b) Every K-bridge in G has at least one simple embedding extending some embedding of K. (Otherwise, its E-graph is a small obstruction and we may stop.) In particular, if some bridge is attached only to two vertices of K, its E-graph is just a branch.
(c) There are no local bridges.
(d) Multiple branches between the same pair of vertices of K have been replaced by a single one.
(e) There are at most $4\,\mathrm{bsize}(K)$ strongly attached bridges. (Otherwise we get an obstruction of bounded size by Lemma 3.1.)

We shall refer to the above assumptions (a)–(e) as *property (E)* of K.

Let $\Xi = (G, K, \Pi, \Delta)$ be a simple EEP where K has property (E). We shall now consider some special subproblems of Ξ. Suppose that $\mathcal{B}$ is a set of K-bridges and $\Xi' = (G, K \cup \mathcal{B}, \Pi', \Delta')$ is a subproblem of Ξ. Then Ξ' is 2-*restricted* if every K-bridge B in G, $B \notin \mathcal{B}$, has at most two Δ'-compatible embeddings extending the embedding Π'.

Suppose that we have a set of vertices $W_0 \subseteq V(K)$. Let W_1 be the union of W_0 and all main vertices of K. Denote by $\mathcal{S}$ the set of connected components of $K - W_1$. Suppose that we replace the paths in $\mathcal{S}$ by new pairwise disjoint paths in $G - W_1$ joining the same ends as the original paths. Then the new subgraph K' of G is homeomorphic to K and the homeomorphism $K \to K'$ is the identity on the stars of vertices in W_1. The types of bridges with respect to K and K' are in the obvious correspondence and so are the embeddings of K and K' and the embedding schemes for their bridges. Suppose that G contains exactly the same types of K-bridges and K'-bridges. Then the replacement of K by K' is called a *compression* with respect to W_0.

THEOREM 5.1 (Juvan and Mohar [20]). *There is a function* $\mathrm{c}_1 : \boldsymbol{N} \times \boldsymbol{N} \to \boldsymbol{N}$ *such that the following holds. Let* $\Xi = (G, K, \Pi, \Delta)$ *be a* 2*-restricted subproblem of an EEP, and let* W_0 *be a set of vertices of* K*. If there is no* Δ*-compatible EE, then there is a compression* $K \mapsto K'$ *with respect to* W_0 *such that the modified EEP* $\Xi' = (G, K', \Pi, \Delta)$ *admits an obstruction* $\mathcal{B}$ *such that* $\mathrm{bsize}_{K'}(\mathcal{B}) \le \mathrm{c}_1(|W_0|, \mathrm{bsize}(K))$*. Moreover, there is an algorithm with time complexity* $O(\mathrm{c}_1(|W_0|, \mathrm{bsize}(K))\,|V(G)|)$ *that either finds an EE for* Ξ *or performs the compression* $K \mapsto K'$ *and returns an obstruction* $\mathcal{B}$ *for* Ξ' *as described above.*

The compression combined with the procedure of embedding distribution of types will be our main tool that will be used in order to guarantee that the obstructions constructed by our algorithms are not too large.

There is another important special instance of EEPs. Suppose that K has property (E) and that there is a Π-face F that contains two singular branches e and f. Suppose that $F = AeBfCe^-Df^-$, where e^- and f^- denote the traversal of e and f, respectively, in the opposite direction and where A, B, C, D are open segments of F between the appearances of e and f. Let $\mathcal{B}$ be a set of K-bridges in G, each of which has an attachment in the interior of e or f. Suppose also that Δ is a simple embedding distribution for bridges in $\mathcal{B}$ such that for each of the types, the embedding schemes allow all together at most one appearance of each basic piece distinct from e, f. Then the EEP $\Xi = (K \cup \mathcal{B}, K, \Pi, \Delta)$ and every EE subproblem of Ξ, $\Xi_0 = (K \cup \mathcal{B}, K \cup \mathcal{B}_0, \Pi_0, \Delta_0)$ ($\mathcal{B}_0 \subseteq \mathcal{B}$), is a *corner EEP*. Every bridge in $\mathcal{B}$ is attached to the interior of e or f. Therefore, under any EE for Ξ (or Ξ_0), all bridges from $\mathcal{B}$ are embedded in the face F. The following nontrivial result has been proved by Marinček, Juvan, and Mohar [18].

THEOREM 5.2 (Juvan, Marinček, and Mohar [18]). *There is a constant* c_0 *such that every corner EEP is the union of at most* c_0 *2-restricted EE subproblems.*

The difficult part of the proof of Theorem 5.2 consists of showing that $\mathcal{B}$ contains a subset $\mathcal{B}_0$ of at most 30 bridges such that for every Δ-embedding of $\mathcal{B}_0$ in F that gives rise to a subface F' of F, which contains a singular segment of e and a singular segment of f, the following holds. For one of the singular branches of F', say ε, the bridges $\mathcal{B}_\varepsilon \subseteq \mathcal{B} \backslash \mathcal{B}_0$ that are attached to ε admit a Δ-embedding (extending the embedding of $\mathcal{B}_0$) such that no Δ-embedding of any of the remaining bridges from $\mathcal{B}$ is obstructed by this embedding. Consequently, the subproblem with such an embedding of $\mathcal{B}_0$ is equivalent to a subproblem where the bridges from $\mathcal{B}_\varepsilon$ have the corresponding fixed embedding. Under this subproblem, F' can be considered as not having two singular branches. Therefore we say that $\mathcal{B}_0$ *removes the double* $\{e, f\}$*-singularity*. Having $\mathcal{B}_0$ with the above property, one can see that each subproblem with a fixed Δ-embedding of $\mathcal{B}_0$ is the union of 2-restricted subproblems. It is shown in [18] that $\mathcal{B}_0$ and additional representatives for further reductions to 2-restricted EEPs can be obtained in linear time. Applying the generalized procedure of embedding distribution of types with compression yields the following theorem.

THEOREM 5.3. *There is a function* $c_2 : \boldsymbol{N} \to \boldsymbol{N}$ *such that the following holds. Let* $\Xi = (G, K, \Pi, \Delta)$ *be a corner EEP with corresponding singular branches* e *and* f*, and let* W_0 *be a set of vertices of* K*. If there is no* Δ*-compatible EE, then there is a compression* $K \mapsto K'$ *with respect to* W_0 *such that the modified corner EEP* $\Xi' = (G, K', \Pi, \Delta)$ *admits an obstruction* $\mathcal{B}$ *of bounded size,* $\mathrm{bsize}_{K'}(\mathcal{B}) \le c_2(|W_0|)$*. Moreover, there is an algorithm with time complexity* $O(c_2(|W_0|)\,|V(G)|)$ *that either finds an EE for* Ξ *or performs a compression* $K \mapsto K'$ *(by changing only segments of* e *and* f*) and returns an obstruction* $\mathcal{B}$ *for* Ξ' *as described above.*

Proof. By Theorem 5.2, Ξ is the union of a bounded number of 2-restricted subproblems $\Xi_i = (G, K \cup \mathcal{B}_0, \Pi_i, \Delta_i)$, $1 \le i \le s$. Moreover, as shown in [18], $\mathcal{B}_0$ and the corresponding subproblems Ξ_i can be generated in linear time, and by using compression with respect to W_0, also the size of $\mathcal{B}_0$ is bounded by certain constant. Let W_1 be the union of W_0 and the set of vertices of attachment of all bridges in $\mathcal{B}_0$. For $i = 1, \ldots, s$, we solve the 2-restricted subproblem Ξ_i by using Theorem 5.1 and perform compression with respect to W_i. Obtaining an EE we stop. Otherwise, let $\mathcal{B}_i$ be the resulting obstruction (of bounded size). It may happen that after the compression $K \mapsto K'$, some K'-bridges in G become large. Therefore, we apply the procedure from [24] in order that K' and its bridges satisfy property (E). We then define W_{i+1} as the union of W_i and vertices of attachment of all bridges in $\mathcal{B}_i$. This choice guarantees that the compression at the ith step does not change any of the previous obstructions $\mathcal{B}_j$ $(j < i)$ and that $\mathcal{B}_j$ remains an obstruction for Ξ_j although the subgraph K has been changed. One can think of a corner EEP as being an embedding into the torus of a graph homeomorphic to K_4. Since $\mathrm{bsize}(K_4) = 6$, Theorem 5.1 implies that the size of $\mathcal{B}_i$ is bounded by $c_1(|W_i|, 6)$. Since s is bounded by the constant c_0 from Theorem 5.2, it follows that $|W_i|$ and $\mathrm{bsize}_{K'}(\mathcal{B}_i)$ are bounded for each i.

After s steps either we find an EE or we stop with a compressed graph K' and the corresponding obstruction $\mathcal{B}' = \mathcal{B}_0 \cup \mathcal{B}_1 \cup \cdots \cup \mathcal{B}_s$ for Ξ' composed of E-graphs of K'-bridges in G. □

Suppose that we have an EEP $\Xi = (G, K, \Pi, \Delta)$, and that $\mathcal{B}$ is an obstruction for all EEPs $\Xi' = (G, K, \Pi, \Delta')$ for which $\Delta' \prec \Delta$. Consider all possible Δ-compatible embedding extensions of Π to $K \cup \mathcal{B}$. Then Ξ is the union of subproblems, in each of which $\mathcal{B}$ has a fixed embedding. In each of these subproblems, for every type T of K-bridges and each embedding scheme $\delta \in \Delta(T)$, there is a bridge of type T in $\mathcal{B}$ that is δ-embedded since otherwise, the embedding of $\mathcal{B}$ would be Δ'-compatible for some $\Delta' \prec \Delta$. Such a bridge is called a *representative* for δ (with respect to the chosen subproblem), and we say that $\mathcal{B}$ is a *complete set of representatives* for Ξ.

The next result will enable us to apply Theorems 5.1 and 5.3 in solving general simple EEPs.

THEOREM 5.4. *Let K be a subgraph of G with property* (E). *Let $\Xi = (G, K, \Pi, \Delta)$ be a simple EEP and suppose that no edge of K appears on a Π-facial walk twice in the same direction. Suppose that $\mathcal{B}_0$ is a complete set of representatives for Ξ and that $K \cup \mathcal{B}_0$ also has property* (E). *Then there is a number c_3 depending only on $\mathrm{bsize}(K \cup \mathcal{B}_0)$ such that each subproblem $\Xi_0 = (G, K \cup \mathcal{B}_0, \Pi_0, \Delta_0)$ of Ξ is equivalent to the union of at most c_3 EE subproblems, each of which is the intersection of a 2-restricted EEP and at most $\mathrm{bsize}(K)/2$ corner EEPs. The decompositions of Ξ_0 to subproblems and of these to corresponding partial problems can be performed in $O(c_3|V(G)|)$ time.*

Proof. Let $\mathcal{B}_0'$ be the set of K-bridges consisting of $\mathcal{B}_0$, all strongly attached $(K \cup \mathcal{B}_0)$-bridges, and all bridges $\mathcal{B}^\circ_{x,y}$, where x, y are arbitrary basic pieces of $K \cup \mathcal{B}_0$, and bridges $\mathcal{B}^\circ_{x,y}$ are defined before Lemma 4.2. Since $K \cup \mathcal{B}_0$ has property (E), the size of $\mathcal{B}_0'$ is bounded by a function of $\mathrm{bsize}(K \cup \mathcal{B}_0)$. Lemma 4.2 implies that Ξ_0 is the union of subproblems $\Xi' = (G, K \cup \mathcal{B}_0', \Pi', \Delta')$ taken over all Δ_0-embeddings of $\mathcal{B}_0' \backslash \mathcal{B}_0$ extending the embedding Π_0 where every 2-restricted type of $(K \cup \mathcal{B}_0)$-bridges in Ξ_0 has its representatives for embedding schemes in Δ'. It suffices to see that every such subproblem Ξ' is the union of a bounded number of subproblems, each of which is equivalent to the intersection of a 2-restricted EEP and at most $\mathrm{bsize}(K)/2$ corner problems.

First, we shall prove that Ξ' is equivalent to the union of a bounded number of subproblems of the form $\Xi'' = (G, K \cup \mathcal{B}_0'', \Pi'', \Delta'')$, where $\mathcal{B}_0''$ consists of $\mathcal{B}_0'$ and some additional bridges. The number of these additional bridges is bounded (depending on $\mathrm{bsize}(K)$).

Recall that $\mathcal{B}_0'$ contains all strongly attached $(K \cup \mathcal{B}_0)$-bridges in G. Because of property (E), $\mathcal{B}_0'$ contains all $(K \cup \mathcal{B}_0)$-bridges that are attached to two main vertices of K. Let $B \notin \mathcal{B}_0'$ be a K-bridge of type $T = \{e, v\}$, where e is an open branch and

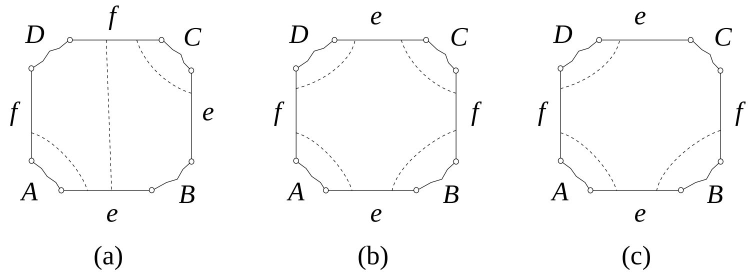


FIG. 5.1. *The possibilities for more than two embedding schemes.*

v is a main vertex of K. Let e_B be the smallest closed segment of e containing all vertices of attachment of B to e. Suppose that F is a Π'-face in which B can be Δ'-embedded. Since B is not a strongly attached $(K \cup \mathcal{B}_0)$-bridge, e_B is contained in an open branch $e' \subseteq e$ of $K \cup \mathcal{B}_0$. Denote by ε an appearance of e' in F. Let $v_1, \ldots, v_l$ be the appearances of v on F. Since $\mathcal{B}_0$ is a complete set of representatives, $\Delta'(T)$ contains at most two embedding schemes using ε and one of $v_1, \ldots, v_l$. Moreover, any embedding extension of Π' to a subset of K-bridges in G can be changed so that all bridges of type T in F that are attached to ε are attached just to one appearance of v in F. This implies that Ξ' is equivalent to a subproblem $\Xi'' = (G, K \cup \mathcal{B}'_0, \Pi', \Delta'')$, where K-bridges that are not attached to two open branches of K have at most two admissible embeddings.

It remains to be seen how we control embeddings of K-bridges that are attached to two open branches of K. For most pairs e, f of open branches, K-bridges of type $\{e, f\}$ will have at most two Δ'-embeddings. This may not be the case only when segments of both e and f appear twice on the same Π'-facial walk. Possible cases are shown in Figure 5.1 with dotted curves indicating the embedding schemes in Δ' that contain appearances of e or f. By assumption, each of the branches e and f appear on the facial walk once in each direction. Therefore we can speak about the left and right side of e and the top or bottom of f (with respect to the presentation in Figure 5.1). We shall assume that the face F shown in Figure 5.1 is a Π-face, and we shall have in mind that there is a collection of K-bridges from $\mathcal{B}'_0$ that are Π'-embedded in F but not explicitly shown.

Let us first consider pairs $T = \{e, f\}$ which correspond to case (a) of Figure 5.1. In each of such cases we shall either conclude that bridges of type T admit at most two Δ''-embeddings (possibly after restricting to an equivalent subproblem), or we will find a bridge B whose presence in $\mathcal{B}''_0$ would guarantee the same as in the former possibility. Since there are only a bounded number of pairs T, we can afterward add all such bridges B to $\mathcal{B}''_0$ and then start again from the beginning. The presence of the added bridges in $\mathcal{B}''_0$ will now guarantee that the former possibility always occurs.

Let $B_1 = B^{i,j}_{e,f}$ $(i, j \in \{1, 2\})$ be the K-bridge corresponding to the rightmost attachment e_i on e and the topmost attachment f_j on f. Note that $B_1 \in \mathcal{B}_0 \cup \mathcal{B}^\circ_{x,y} \subseteq \mathcal{B}'_0$ for some $x \subseteq e$, $y \subseteq f$. Assume first that B_1 is Π'-embedded in F so that it is attached to the right occurrence of e. Then B_1 is attached to f at its upper occurrence since the other possibility is not Δ''-compatible. Let y be an attachment of B_1 to f. By the choice of B_1, if y is not the only attachment of B_1 on f, then each of the bridges of type T admits at most two Δ''-embeddings extending Π', and we are done. If y is the only attachment, then we can have bridges of type T with three distinct Δ''-embeddings. However, the set $\mathcal{B}'$ of such bridges has only one attachment on f;

it is equal to y. (Another possibility for bridges with three embeddings in F includes bridges of type T whose only attachment on e is e_i. Though, this case is excluded since the left-right embeddings in F are not Δ''-compatible.) The two occurrences of y on F separate F into two segments. If no bridge from $\mathcal{B}'_0$ is embedded in F such that it is attached to the interior of each of these segments, then every EE of Π' to a subset of K-bridges can be changed so that no bridge from $\mathcal{B}'$ is attached to the left occurrence of y (say). In other words, Ξ'' is equivalent to a subproblem where each bridge of type T has only two allowed embeddings (and we shall assume that this subproblem is already Ξ''). On the other hand, if there is a Π'-embedded bridge $B_2 \in \mathcal{B}'_0$ in F that separates the two occurrences of y, there is only one possibility for a bridge of type T to have three possible Δ''-embeddings. Such a bridge B must be attached only to two vertices, and so it is just a branch by property (E). In this case we shall add B in $\mathcal{B}''_0$. Then we will be able to forget about B having three distinct embeddings on the expense of a few additional subproblems to be solved.

The second possibility is when B_1 is attached to the lower occurrence of e and the left occurrence of f. Now, the only bridges of type T with more than two possible Δ''-embeddings have their only vertex on e equal to e_i. We conclude in the same way as we did in the first case, using e_i instead of y.

The third possibility is when B_1 is embedded so that it is attached to the lower and the upper occurrence of e and f, respectively. In this case, there are two ways that bridges could have more than two Δ''-embeddings extending Π'. If $B^{i,3-j}_{e,f} \neq B_1$, then one of the two possibilities is excluded. The remaining one is essentially the same as the second possibility treated above. On the other hand, if $B^{i,3-j}_{e,f} = B_1$, then we have a situation that is essentially the same as the first case above. In each case we know how to act.

Let us now consider cases (b) and (c) of Figure 5.1. In case (c), it may happen that there is an embedding scheme in Δ'' containing an appearance of a basic piece in the segment C and the left occurrence of f (or the bottom occurrence of e). In such a case, bridges of type $\{e, f\}$ may be assumed to have only two possible embeddings. This is established in the same way as above (by possibly adding a new bridge to $\mathcal{B}''_0$ or restricting to an equivalent subproblem). We assume from now on that this is not the case.

We say that $\mathcal{B}''_0$ *removes the double* $\{e, f\}$*-singularity* if no subface F' of F contains singular branches $e' \subseteq e$ and $f' \subseteq f$ such that there exist $(K \cup \mathcal{B}''_0)$-bridges attached to each of e' and f'. If the Π''-embedded bridges $\mathcal{B}''_0$ do not remove the double $\{e, f\}$-singularity, then $\{e, f\}$ is a *corner pair* for Ξ''. Since $\mathcal{B}_0$ is a complete set of representatives, distinct corner pairs are disjoint. Therefore there are at most bsize$(K)/2$ corner pairs. If case (b) or (c) applies for $T = \{e, f\}$ and T is not a corner pair, then reductions from [18] (by possibly extending $\mathcal{B}''_0$ or restricting Δ'' to an equivalent subproblem) can be used to get subproblems where all K-bridges of type T have at most two Δ''-embeddings. This will be assumed in the sequel as already done.

If $\{e, f\}$ is a corner pair, let $\mathcal{B}^{e,f}_1$ be the set of K-bridges in G of type $\{e, f\}$ that are not in $\mathcal{B}''_0$. Let $\mathcal{B}_2$ be the set of K-bridges that are not in $\mathcal{B}''_0$ and that are not in $\mathcal{B}^{e,f}_1$ for any corner pair $\{e, f\}$. Furthermore, let $\mathcal{B}^{e,f}_2$ contain all K-bridges from $\mathcal{B}_2$ that have an attachment on e or f and have at most one Δ''-embedding extending the embedding Π'' of $K \cup \mathcal{B}''_0$. Similarly, let $\mathcal{B}_1$ contain those bridges from $\mathcal{B}^{e,f}_1$, taken over all corner pairs $\{e, f\}$, which have at most one Δ''-embedding extending Π''.

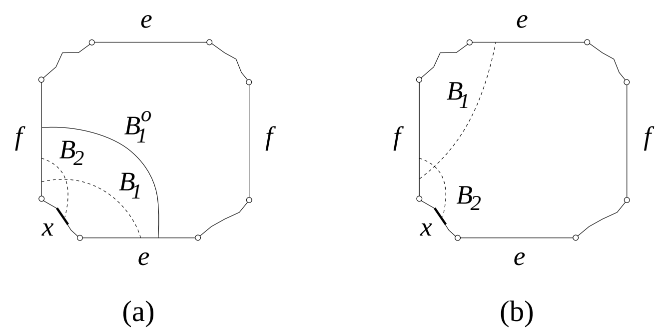


FIG. 5.2. *B_1 and B_2 overlap.*

Consider the EEPs

$$\Xi_1^{e,f} = (K \cup \mathcal{B}_0'' \cup \mathcal{B}_1^{e,f} \cup \mathcal{B}_2^{e,f}, K \cup \mathcal{B}_0'', \Pi'', \Delta_1^{e,f}), \tag{5.1}$$

where $\{e, f\}$ is a corner pair and $\Delta_1^{e,f}$ is the restriction of Δ'' to $\mathcal{B}_1^{e,f} \cup \mathcal{B}_2^{e,f}$. Let

$$\Xi_2 = (K \cup \mathcal{B}_0'' \cup \mathcal{B}_1 \cup \mathcal{B}_2, K \cup \mathcal{B}_0'', \Pi'', \Delta_2) \tag{5.2}$$

be the partial problem of Ξ'' restricted to $\mathcal{B}_1 \cup \mathcal{B}_2$. We claim that Ξ'' is the intersection of partial problems $\Xi_1^{e,f}$ (taken over all corner pairs $\{e, f\}$) and Ξ_2. Suppose not. Since different corner pairs do not obstruct each other, there is an EE Π_1 for some $\Xi_1^{e,f}$ and an EE Π_2 for Ξ_2 that cannot be combined into an EE for Ξ''. This means that a Π_1-embedded bridge $B_1 \in \mathcal{B}_1^{e,f} \backslash \mathcal{B}_1$ and a Π_2-embedded bridge $B_2 \in \mathcal{B}_2 \backslash \mathcal{B}_2^{e,f}$ overlap. Since $\mathcal{B}_0$ is a complete set of representatives, B_1 overlaps only with bridges that are attached to e or to f. Since $B_2 \notin \mathcal{B}_0''$, we may assume that B_2 is of type $\{f, x\}$ where $x \subseteq A$. See Figure 5.2 where the cases (a) and (b) from below are distinguished.

An embedding of $B \in \mathcal{B}_1^{e,f}$ is an *embedding in the corner* α if B is attached to the lower occurrence of e and the left occurrence of f. Similarly, we define embeddings in corners β, γ, δ as those that are using the lower/right, upper/right, or upper/left occurrences of e/f, respectively. In the obvious way we also classify embeddings of bridges of type $\{f, x\}$ to be in corners α, β, γ, or δ. We may assume that B_2 is Π_2-embedded in the corner α.

Since $B_1 \notin \mathcal{B}_0''$, there is a $(K \cup \mathcal{B}_0)$-bridge $\tilde{B}_1 \in \mathcal{B}_0''$ that is of the same type $\{e_1, f_1\}$, $e_1 \subseteq e$, $f_1 \subseteq f$, as B_1 and of the form $B_{f_1,e_1}^{i,j}$, where $(f_1)_i$ refers to the lowest attachment on f_1. Similarly, there is $\tilde{B}_2 \in \mathcal{B}_0''$ of the same type $\{e_2, f_2\}$ as B_2 and of the form $B_{f_2,e_2}^{k,l}$, where $(f_2)_k$ refers to the topmost attachment on f_2. Now we distinguish two cases.

Case (a). *Embeddings of bridges of type $\{e, f\}$ in corner α are $\Delta_1^{e,f}$-compatible.* There is a representative $B_1^\circ \in \mathcal{B}_0$ that is Π''-embedded (and hence also Π_1-embedded and Π_2-embedded) in the corner α. Therefore B_1 is Π_1-embedded in α as well. Since $\tilde{B}_1 \in \mathcal{B}_0''$, it does not overlap with B_2 under the embedding Π_2. Since $\{e, f\}$ is a corner pair, $\tilde{B}_1$ and B_1° do not remove the double $\{e, f\}$-singularity. Hence $\tilde{B}_1$ is embedded in the corner β. Consequently, $\tilde{B}_2$ is not Π''-embedded in β. If $\tilde{B}_2$ is embedded in the corner γ or δ, B_1° and $\tilde{B}_2$ remove the double $\{e, f\}$-singularity. Hence, $\tilde{B}_2$ is Π''-embedded in α. This implies that B_1 cannot be embedded in the corner α, a contradiction.

Case (b). *Embeddings in corner α are not $\Delta_1^{e,f}$-compatible.* Let $B_\beta, B_\gamma, B_\delta$ be representatives from $\mathcal{B}_0$ that are Π''-embedded in corners β, γ, δ, respectively. We shall distinguish four cases according to where $\tilde{B}_2$ is Π''-embedded.

Subcase α. $\tilde{B}_2$ is in the corner α. This contradicts the fact that B_1 is Π_1-embedded in α.

Subcase β. $\tilde{B}_2$ being in β, $\tilde{B}_1$ is in α or in δ. Since embeddings in α are not $\Delta_1^{e,f}$-compatible, $\tilde{B}_1$ is in δ. This eliminates the possibility for B_2 being Π_2-embedded in the corner α, and we are done.

Subcase γ. $\tilde{B}_2$ is in γ. Denote by y the lowest attachment of B_γ to f. When adding the sets $\mathcal{B}^\circ_{x',y'}$ ($x' \subseteq x, y' \subseteq f$) into $\mathcal{B}'_0 \subseteq \mathcal{B}''_0$ and restricting Δ'' according to Lemma 4.2 we have ensured that there are representatives for all embedding schemes of $(K \cup \mathcal{B}_0)$-bridges of such types $\{x', y'\}$. Since $\{e, f\}$ is a corner pair for Ξ'', such bridges with embeddings in corner α have all their attachments to f strictly below y. Therefore, they all belong to $\mathcal{B}_2^{e,f}$. In particular, this holds for the bridge B_2, and we have a contradiction.

Subcase δ. $\tilde{B}_2$ is in δ. Let y be the lowest attachment of B_δ to f. We conclude as above.

This proves that Ξ'' is the intersection of corner problems $\Xi_1^{e,f}$ and Ξ_2. Let us observe that the ≤ 2 embeddings of bridges from $\mathcal{B}_2$ are determined by their types as $(K \cup \mathcal{B}''_0)$-bridges. Therefore, Ξ_2 can be formulated as a 2-restricted EEP, and the proof is complete. ∎

The assumption in Theorem 5.4 that no edge appears on a Π-facial walk twice in the same direction is not essential. We have decided to use it since it eliminates a few cases in the proof and since this condition will be automatically satisfied at the time when applying the theorem. Let us also mention that with a slightly modified proof of Theorem 5.4, one can achieve c_3 being bounded only by a function of $\mathrm{bsize}(K)$.

Corollary 5.5. *Let* $\Xi = (G, K, \Pi, \Delta)$ *be a simple EEP and let* W_0 *be a subset of vertices of* K*. Suppose that* K *has property* (E) *and that no edge of* K *appears on a* Π*-facial walk twice in the same direction. There is a function* $\mathrm{c} : \boldsymbol{N} \times \boldsymbol{N} \to \boldsymbol{N}$ *and an algorithm with time complexity* $O(\mathrm{c}(|W_0|, \mathrm{ord}(\Delta))|V(G)|)$ *that either finds a* Δ*-compatible EE or returns a subgraph* K' *of* G *obtained by a compression with respect to* W_0 *and a set of at most* $\mathrm{c}(|W_0|, \mathrm{ord}(\Delta))$ *E-graphs of* K'*-bridges in* G *that form an obstruction for the corresponding EEP* $\Xi' = (G, K', \Pi, \Delta)$.

Proof. The proof is by induction on $\mathrm{ord}(\Delta)$. If $\mathrm{ord}(\Delta) = 0$, then any K-bridge in G is an obstruction for Ξ. Hence, a Δ-embedding exists if and only if $K = G$. Suppose now that $\mathrm{ord}(\Delta) > 0$. There are $\mathrm{ord}(\Delta)$ embedding distributions $\Delta_1, \Delta_2, \ldots$ that are strictly simpler than Δ and are maximal with this property. Inductively, we first solve the subproblem $\Xi_1 = (G, K, \Pi, \Delta_1)$ taking care of the set W_0. An EE makes us happy and we stop. Otherwise, we compress K with respect to W_0. Let K_1 be the new subgraph of G and $\mathcal{B}_1$ an obstruction of bounded size as guaranteed by the induction hypothesis. Let W_1 be the union of W_0 and the set of vertices of attachment of bridges from $\mathcal{B}_1$. Now we replace W_0 by W_1 and solve the subproblem $\Xi_2 = (G, K_1, \Pi, \Delta_2)$, taking care of the set W_1. Either we stop, or we get a new graph K_2 (after a compression with respect to W_1) and an obstruction $\mathcal{B}_2$ of bounded size. In the latter case we extend W_1 into W_2 by adding all attachments of bridges from $\mathcal{B}_2$. Continuing, we either find an EE, which is a Δ-embedding as well, or we stop after $\mathrm{ord}(\Delta)$ steps with a subgraph K' of K that is a compression of K with respect to W_0. At the same time we get an obstruction $\mathcal{B}_0 = \mathcal{B}_1 \cup \mathcal{B}_2 \cup \cdots$. Now, since $\mathcal{B}_0$ is an obstruction for all simpler EEPs, it is a complete set of representatives for Ξ. Since Ξ is the union of subproblems, taken over all Δ-embeddings of $\mathcal{B}_0$, and since $\mathcal{B}_0$ has bounded size, we can consecutively apply Theorem 5.4 combined with Theorems 5.1 and 5.3 and for each of these subproblems perform a compression with respect to attachments of E-graphs in all previously obtained obstructions. An upper bound on $c(|W_0|, \mathrm{ord}(\Delta))$ is easy to obtain by our inductive approach, and we leave the details to the reader. □

6. Embedding graphs in an arbitrary surface. In this section we prove the final result of this paper that embeddability in any fixed surface S can be decided in linear time. Our algorithm not only verifies if such an embedding exists. If it does, such an embedding is constructed. If not, the algorithm identifies a subgraph of G that cannot be embedded in S but every proper subgraph can. Such a subgraph is called a *minimal forbidden subgraph* for embeddability in S. We define the *Euler genus* of S as $2 - \chi(S)$, where $\chi(S)$ is the Euler characteristic of S.

THEOREM 6.1. *Let S be a fixed closed surface. There is a constant c and a linear time algorithm that for an arbitrary given graph G either*

(a) *finds an embedding of G in S, or*
(b) *identifies a minimal forbidden subgraph $K \subseteq G$ for embeddability in S. The branch size of K is bounded by c.*

Remark. In case (a), our algorithm constructs an embedding in the surface of the smallest Euler genus (and the same orientability characteristic as S). Such an embedding determines a (possibly not 2-cell) embedding in S. If one insists on 2-cell embeddings in S, there is a polynomial time solution using an algorithm for the maximum genus [12] (which turns out to be trivial for nonorientable surfaces, cf., e.g., [26]).

A corollary of Theorem 6.1 is the result of Robertson and Seymour [27] that the set of minimal forbidden minors (or subgraphs) is finite for each surface. It is worth mentioning that our proof is constructive while the proof in [27] is only existential.

COROLLARY 6.2 (Robertson and Seymour [27]). *For every surface S there is a finite list of graphs such that an arbitrary graph G can be embedded in S if and only if G does not contain a subgraph homeomorphic to one of the graphs in the list.*

The rest of the paper is devoted to the proof of Theorem 6.1. Let us just point out that in case (b) it suffices to find a subgraph K of bounded branch size (in terms of the Euler genus of S) since such a subgraph is easily changed to a minimal one in constant time (for example, by considering all subgraphs of K, up to homeomorphism, and all their embeddings).

Denote by g the Euler genus of S. If S is orientable, our algorithm determines the smallest $h \leq g$ such that G can be embedded in the orientable surface of Euler genus h (or proves that such an h does not exist). If S is nonorientable, then we will determine the surface (or two surfaces) with the smallest Euler genus $h \leq g$ in which G can be embedded (or show that G cannot be embedded in S). If such minimal Euler genus h is even, there is a nonorientable surface $\tilde{S}_h$ as well as an orientable surface S'_h with Euler genus h. If G can be embedded in $\tilde{S}_h$ and $h \leq g$, then it can also be embedded in S. If G has an embedding in S'_h, then changing the sign of an arbitrary edge which is not a cutedge of G gives an embedding in $\tilde{S}_{h+1}$. Hence, any outcome determines the nonorientable genus of G.

The orientable genus of G is equal to the sum of the genera of its blocks [4] and a minimum genus embedding is a simple combination of minimal embeddings of the blocks. A similar reduction works in the nonorientable case [33]. Since the blocks can be determined in linear time, we may assume from now on that the graph G is 2-connected.

If G is 2-connected and $G = G_1 \cup G_2$, where $G_1 \cap G_2 = \{x, y\} \subseteq V(G)$ and each of G_1 and G_2 contains at least two edges, then we say that $\{x, y\}$ is a *separating pair*.

In such a case, let the graph G_i' be obtained from G_i by adding the edge xy if it is not already present in G_i $(i = 1, 2)$. The added edge xy is called the *virtual edge* of G_i'. If G_1' is 3-connected, then G_1' is a *pendant* 3-*connected block* of G. If G_2' is planar, then every embedding of G_1' can be changed into an embedding of G in the same surface after replacing the edge $xy \in E(G_1')$ by G_2 using a planar embedding of G_2'. In such a case we call the operation of replacing G by G_1' a 2-*reduction*. We can consider the graph G_1' as being a subgraph of G by using a path in G_2 from x to y instead of the new edge xy. Therefore, any obstructions in G_1' give rise to obstructions of the same branch size in G. By using linear time algorithms of Hopcroft and Tarjan to determine the 3-connected components of G [15] and for testing planarity [16], we can perform all possible 2-reductions in linear time. At the same time we locate all pendant 3-connected blocks in G, and for each such block B we find a Kuratowski subgraph $H_B \subseteq B$. If possible, we choose H_B so that it does not contain the virtual edge of B.

We shall assume from now on that G is a 2-connected graph in which no 2-reductions are possible. In particular, G is simple and has no vertices of degree 2. The following lemmas will be used to bound the number of pendant 3-connected blocks.

LEMMA 6.3. *Suppose that $K = L \cup H$, where H is a subgraph of K homeomorphic to a Kuratowski graph and that $L \cap H$ is either empty, one or two vertices, a segment of a branch of H, or a segment of a branch of L. If g' is the Euler genus of K and g is the Euler genus of L, then $g' \geq g + 1$.*

Proof. By the additivity of the Euler genus, the result is clear when $L \cap H$ is empty or a single vertex. Otherwise, let x and y be the two vertices of of $L \cap H$ or the ends of the segment of a branch (of L or of H) in $L \cap H$, respectively. Since $L \subseteq K$, we have $g \leq g'$.

If $g' = g$, consider an embedding Π' of K with Euler genus g. It is an extension of an embedding Π of L. Since Π is an embedding of L of minimal Euler genus, no Π-facial walk W contains two vertices that appear on W in the interlaced order. (If not, one could change Π to an embedding with smaller Euler genus.) This immediately excludes the case when $L \cap H = \{x, y\}$. Similarly, if $L \cap H$ is contained in a branch of H, then since K_5 and $K_{3,3}$ are 3-connected, there is a single L-bridge in K. It is attached to x and y only, and it does not have a simple EE. Therefore, x and y appear interchangeably on a Π-facial walk, a contradiction. The remaining case is when $L \cap H$ is a segment σ of a branch e of L. Let $L' = L - \operatorname{int} \sigma$. Since K_5 and $K_{3,3}$ are 3-connected, there are one or two L'-bridges in K. In the latter case, one of the L'-bridges is just a segment of a branch of H, and by replacing σ with that branch we can appeal to the previous case treated above. So, we may assume that there is a single L'-bridge in K; it is equal to H. If the branch e is contained in two Π-facial walks, the embedding extension of $\Pi|L'$ to $L' \cup H = K$ gives a contradiction as above. On the other hand, if e is singular, it appears on the facial walk twice in opposite direction and hence the embedding of K yields an embedding of H in the cylinder, a contradiction. □

LEMMA 6.4. *Let G be a 2-connected graph. Suppose that $G = K \cup B_1^- \cup \cdots \cup B_s^-$ $(s \geq 5)$, where K has a branch e containing vertices $x_1, y_1, x_2, y_2, \ldots, x_s, y_s$ (in that order; $x_i \neq y_i$ but possibly $y_i = x_{i+1}$) such that $K \cap B_i^-$ is equal to the segment of e from x_i to y_i $(1 \leq i \leq s)$, and if $B_i^- \cap B_{i+1}^- \neq \emptyset$ for some $1 \leq i < s$, then $B_i^- \cap B_{i+1}^- = \{y_i\} = \{x_{i+1}\}$. Suppose, moreover, that each of the graphs B_i^- is planar but $B_i^- + x_i y_i$ is nonplanar $(1 \leq i \leq s)$. Let $G' = K \cup B_1^- \cup B_3^- \cup B_5^-$. If Π' is a* *minimum genus or a minimum Euler genus embedding of G' in the surface S, then Π' can be changed into an embedding of G in S.*

Proof. Since $B_i^- + x_i y_i$ is not planar and since G (and hence also K) contains a path from x_i to y_i that is edge-disjoint from B_i^-, B_i^- contains a Π'-noncontractible cycle C_i ($i = 1, 3, 5$). The graphs $B_1^-, \dots, B_s^-$ are planar and distinct B_i^-, B_j^- intersect in at most one vertex which belongs to e. This implies that there is an embedding Π_0 of $e \cup B_1^- \cup \cdots \cup B_s^-$ of genus 0. If C_1 is 2-sided, then Π' restricted to K and Π_0 are easily combined into an embedding of G in S (similarly, if C_3 or C_5 is 2-sided). On the other hand, if C_1, C_3, C_5 are all 1-sided, then the Euler genus of Π' restricted to $K - e$ is smaller than the Euler genus of Π' by at least three since C_1, C_3, C_5 are disjoint. Now, the same surgery as used in the 2-sided case yields an embedding of G whose Euler genus increases by at most two. This is a contradiction to minimality of Π'. □

Our next goal is to find a 2-connected subgraph K of G such that no K-bridges in G are local. First we construct an intermediate graph K_0. If G is 3-connected, then we let K_0 be a Kuratowski subgraph of G. Otherwise, for each pendant 3-connected block B of G, let K_B be its subgraph obtained by the following construction. Let H_B be a Kuratowski subgraph of B and let $\{x, y\}$ be the separating pair of G corresponding to B. If H_B contains the virtual edge xy, then put $K_B = H_B - xy$. Otherwise, let K_B be obtained from H_B by adding two disjoint paths (possibly of length 0) from $\{x, y\}$ to H_B. The graphs K_B are easily constructed in linear time by standard techniques mentioned earlier in this paper. Now, we start by taking $K_0 = K_{B_0}$, where B_0 is an arbitrary pendant 3-connected block of G. We shall extend K_0 in several steps. Note that K_0 may become 2-connected only after the next step. In each of these steps we first check if there is a pendant 3-connected block B such that either K_B is edge-disjoint from the current graph, or $K_B \supseteq H_B$. If so, we add K_B and two disjoint paths from its separating set to the current graph. If one of such paths passes through a pendant 3-connected block Q, we make sure that inside Q it uses only edges of K_Q. By Lemma 6.3, the new graph K_0 has larger Euler genus than the previous one, so this case occurs at most g times (or else we get a small forbidden subgraph for embeddability in S and stop). After $O(g)$ such steps, each of the remaining pendant 3-connected blocks B has the property that B^- (B without its virtual edge) is planar and that $K_0 \cap K_B$ is a segment of a branch e of K_0. We say that B is pendant on e.

Consider the bridges $B_1, \dots, B_s$ that are pendant on e, in the order as their segments $K_0 \cap B_i^-$ appear on e. By Lemma 6.4 we may assume that $s \le 4$ (possibly after changing the graph G by replacing B_2, B_4, and $B_6, \dots, B_s$ by corresponding segments of e). Now we add the graphs K_{B_i}, $i = 1, \dots, s$, into K_0. Note that in this case, there is no need to add corresponding linking paths. We repeat the same for all branches e of K_0, and then our construction stops. Since we make all together $O(g)$ steps, we can afford to spend $O(n)$ time for each step; hence there is no problem in achieving linear time complexity in the construction of K_0.

The graph K_0 constructed above is 2-connected and $\mathrm{bsize}(K_0)$ is bounded. For each branch e of K_0, let $\mathrm{local}(e, K_0)$ be the union of e and all local K_0-bridges on e. If $\{x, y\}$ is a separating pair of G, then each component of $G - \{x, y\}$ intersects some pendant 3-connected block and hence contains a main vertex of K_0. This property of K_0 enables us to use a linear time algorithm from [17] to achieve one of the following:

(a) We get a path e' in $\mathrm{local}(e, K_0)$ joining the ends of e such that the graph $K_0' = K_0 - e + e'$ has no local bridges on e'. Note that $\mathrm{local}(f, K_0') = \mathrm{local}(f, K_0)$ for all branches $f \neq e$ of K_0', and that $\mathrm{local}(e', K_0') = e'$.

(b) We get a subgraph $K_e \subseteq \text{local}(e, K_0)$ that is homeomorphic to a Kuratowski graph. In this case we delete e from K_0, and then add K_e and paths in $\text{local}(e, K_0)$ from the ends of e to K_e so that the resulting graph K_0' is 2-connected. Note that this step increases the branch size of the graph at most by 13.

We repeat the procedure with the new graph K_0' and all its branches f for which $\text{local}(f, K_0') \neq f$. Lemma 6.3 shows that after a bounded number of steps we either stop with a 2-connected graph $K \subseteq G$ such that there are no local K-bridges in G (which we assume henceforth), or we find a subgraph of G of bounded branch size that cannot be embedded in S.

Having constructed K as explained above, the algorithm continues by induction on the genus g of S (or the Euler genus g of S if S is nonorientable). Recursively, either we have either found an embedding in a surface of (Euler) genus smaller than g (in which case we stop), or we got a 2-connected subgraph K of G that cannot be embedded in any surface with (Euler) genus smaller than g. By the induction hypothesis (or by the above construction if $g = 0$), $\text{bsize}(K)$ is bounded. Therefore, K has only a bounded number of embeddings in S (and each of them is 2-cell). Existence of an embedding of G in S is thus equivalent to the existence of an EE with respect to a bounded number of EEPs corresponding to particular embeddings of K in S. By solving all these problems (and successively performing compressions, if necessary, and taking care that vertices of attachment of bridges in previously obtained obstructions are not changed during later compressions), we get either an embedding of G in S or the union of obstructions for the EEPs gives a subgraph $\tilde{K}$ of bounded branch size that cannot be embedded in S. If we will use $\tilde{K}$ in further processing, we just make sure that there are no local $\tilde{K}$-bridges. This can be done in the same way as in the construction of the initial subgraph K.

It remains to see how we solve an EEP $\Xi = (G, K, \Pi, \Delta)$, where Δ contains all embedding schemes that are possible under the given embedding Π of K in the surface S. Let us first verify that no edge of K appears on a Π-facial walk F traversed twice in the same direction. This is clear if S is orientable. If S is nonorientable, changing the signature on such an edge would change Π into an embedding with the same facial walks except that F splits into two facial walks. This contradicts the fact that Π is an embedding of K with minimal Euler genus.

We will construct a sequence of graphs $K_0, K_1, \ldots$ such that $K_0 = K$ and K_{i+1} is obtained (after a compression) from K_i by adding an obstruction for simple embedding extensions. Let us describe the construction of K_{i+1} ($i = 0, 1, 2, \ldots$) in more details. First of all, we replace each K_i-bridge in G by its E-graph. This can be done in linear time by Theorem 3.2. By using Corollary 5.5, we get in linear time the set $\mathcal{B}_i$ of K_i-bridges in a compressed obstruction for simple embedding extensions of K_i to G, taken over all EEs of Π to K_i. Of course, having found an EE, we stop and by Theorem 3.2 we also get an EE of K_0 to G. Assuming that no EE has been found, and assuming inductively that the branch size of K_i is bounded, $\text{bsize}_{K_i}(\mathcal{B}_i)$ is also bounded (Corollary 5.5). We now define $K_{i+1} = K_i \cup \mathcal{B}_i$ and observe that there are no K_{i+1}-bridges that are local on a branch of K_{i+1} contained in K_i. On the other hand, bridges that are local on branches from $\mathcal{B}_i$ can be eliminated by the algorithm from [17] similarly to the very beginning of our algorithm. After doing that, we stop if $K_{i+1} = G$ or if K_{i+1} has no embeddings in S.

Note that for each i, $\mathcal{B}_i \neq \emptyset$ (or we stop with an embedding). Therefore, the above process terminates after a finite number of steps. We claim that the num-

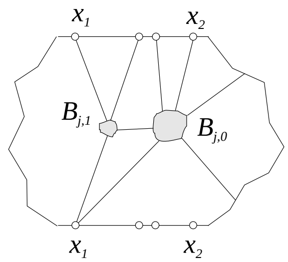


FIG. 6.1. $B_{j,1}$ *does not increase* q.

ber of steps cannot be too large. Let $B_1, \ldots, B_k$ be the K_0-bridges in K_i $(i \geq 1)$. (When constructing K_i, we may have used a compression and thus have changed $K_{i-1}, K_{i-2}, \ldots, K_0$. But a compression is a graph homeomorphism which is identity on the neighborhoods of main vertices of K_{i-1}, and hence we can also view $K_0, K_1, \ldots$ as being subgraphs of the changed graph K_{i-1}.) Since $\mathcal{B}_0, \ldots, \mathcal{B}_{i-1}$ always consist of E-graphs with respect to $K_0, \ldots, K_{i-1}$, respectively, each B_j $(1 \leq j \leq k)$ can be written as $B_j = B_{j,0} \cup B_{j,1} \cup \cdots \cup B_{j,i-1}$, where $B_{j,l} = B_j \cap \mathcal{B}_l$, $l = 0, \ldots, i-1$. Let us consider an embedding Π_i of K_i in S as an EE of the embedding Π of K_0. Then an E-graph in some $B_{j,0}$ is nonsimply embedded. This implies that B_j is attached to at least three appearances of basic pieces of K_0. Consider the sum

$$\sum_{r=1}^{k} (q(B_r) - 2), \tag{6.1}$$

where $q(B_r)$ is defined in Lemma 3.1. Now, $q(B_j)$ contributes at least 1 to (6.1). Let us now consider the induced embedding of Π_i to K_2 as an extension of the embedding of K_1. Since $\mathcal{B}_1 \subseteq K_2$ is an obstruction for simple extensions of K_1, there is an E-graph B in some $B_{j,1}$ that is not simply embedded. We claim that we can choose B such that $q(B_{j,0} \cup B_{j,1}) \geq q(B_{j,0} \cup B) > q(B_{j,0})$ (in all three cases viewed as K_0-bridges). If this is not the case, then B is attached only to $B_{j,0}$ and to the same appearances of basic pieces of K_0 as $B_{j,0}$. No basic piece in $B_{j,0} \backslash K_0$ is singular under the considered embedding of K_1. Hence $B_{j,1}$ is attached to two appearances of a basic piece x' of K_1, and if $x \supseteq x'$ is the basic piece of K_0 containing x', then $B_{j,0}$ is attached to the corresponding appearances of x. Since $B_{j,0}$ is an E-graph of a K_0-bridge, it contains feet at extreme attachments x_1, x_2 of B_j on x. We have shown above that no edge of K_0 appears on a Π-facial walk twice in the same direction. It follows that the embedding of $B \subseteq B_{j,1}$ is as shown in Figure 6.1 and that x' is an extreme attachment of B_j, say $x' = x_1$. However, this embedding can easily be changed so that B is not attached to the upper occurrence of x_1 (say), without affecting possible embeddings of other bridges from $\mathcal{B}_1$. After doing the same with other candidates for B, we get a contradiction with $\mathcal{B}_1$ being an obstruction for simple embeddings.

The same proof can be carried further, for embeddings of K_3, K_4, etc. We conclude that the sum (6.1) is at least i. Now, Lemma 3.1 implies that $i \leq 4\,\mathrm{bsize}(K_0)$. The proof is complete.

REFERENCES

[1] D. ARCHDEACON, *A Kuratowski theorem for the projective plane*, J. Graph Theory, 5 (1981), pp. 243–246.

[2] D. Archdeacon, *The complexity of the graph embedding problem*, in Topics in Combinatorics and Graph Theory, R. Bodendiek and R. Henn, eds., Physica-Verlag, Heidelberg, 1990, pp. 59–64.
[3] D. Archdeacon and P. Huneke, *A Kuratowski theorem for nonorientable surfaces*, J. Combin. Theory Ser. B, 46 (1989), pp. 173–231.
[4] J. Battle, F. Harary, Y. Kodama, and J. W. T. Youngs, *Additivity of the genus of a graph*, Bull. Amer. Math. Soc., 68 (1962), pp. 565–568.
[5] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, North–Holland, New York, 1981.
[6] K. S. Booth and G. S. Lueker, *Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-trees*, J. Comput. System Sci., 13 (1976), pp. 335–379.
[7] N. Chiba, T. Nishizeki, S. Abe, and T. Ozawa, *A linear algorithm for embedding planar graphs using PQ-trees*, J. Comput. System Sci., 30 (1985), pp. 54–76.
[8] S. A. Cook and R. A. Reckhow, *Time bounded random access machines*, J. Comput. Syst. Sci., 7 (1976), pp. 354–375.
[9] H. Djidjev and J. H. Reif, *An efficient algorithm for the genus problem with explicit construction of forbidden subgraphs*, in 23rd Annual ACM Symposium on Theory of Computing, New Orleans, LA, May 1991, pp. 337–347.
[10] I. S. Filotti, G. L. Miller, and J. Reif, *On determining the genus of a graph in* $O(v^{O(g)})$ *steps*, in Proc. 11th Annual ACM STOC, Atlanta, GA, 1979, pp. 27–37.
[11] H. de Fraysseix and P. Rosenstiehl, *A depth-first search characterization of planarity*, Ann. Discrete Math., 13 (1982), pp. 75–80.
[12] M. L. Furst, J. L. Gross, and L. A. McGeoch, *Finding a maximum-genus graph imbedding*, J. Assoc. Comput. Mach., 35 (1988), pp. 523–534.
[13] H. Glover, J. P. Huneke, and C. S. Wang, 103 *graphs that are irreducible for the projective plane*, J. Combin. Theory Ser. B, 27 (1979), pp. 332–370.
[14] J. L. Gross and T. W. Tucker, *Topological Graph Theory*, Wiley-Interscience, New York, 1987.
[15] J. E. Hopcroft and R. E. Tarjan, *Dividing a graph into triconnected components*, SIAM J. Comput., 2 (1973), pp. 135–158.
[16] J. E. Hopcroft and R. E. Tarjan, *Efficient planarity testing*, J. Assoc. Comput. Mach., 21 (1974), pp. 549–568.
[17] M. Juvan, J. Marinček, and B. Mohar, *Elimination of local bridges*, Math. Slovaca, 47 (1997), pp. 85–92.
[18] M. Juvan, J. Marinček, and B. Mohar, *Obstructions for simple embeddings*, submitted.
[19] M. Juvan, J. Marinček, and B. Mohar, *Embedding a graph into the torus in linear time*, submitted.
[20] M. Juvan and B. Mohar, *Extending* 2*-restricted partial embeddings of graphs*, submitted.
[21] A. Karabeg, *Classification and detection of obstructions to planarity*, Linear and Multilinear Algebra, 26 (1990), pp. 15–38.
[22] B. Mohar, *Projective planarity in linear time*, J. Algorithms, 15 (1993), pp. 482–502.
[23] B. Mohar, *Obstructions for the disk and the cylinder embedding extension problems*, Combin. Probab. Comput., 3 (1994), pp. 375–406.
[24] B. Mohar, *Universal obstructions for embedding extension problems*, submitted.
[25] B. Mohar and C. Thomassen, *Graphs on Surfaces*, The Johns Hopkins University Press, Baltimore, MD, to appear.
[26] G. Ringel, *The combinatorial map color theorem*, J. Graph Theory, 1 (1977), pp. 141–155.
[27] N. Robertson and P. D. Seymour, *Graph minors.* VIII. *A Kuratowski theorem for general surfaces*, J. Combin. Theory Ser. B, 48 (1990), pp. 255–288.
[28] N. Robertson and P. D. Seymour, *Graph minors.* XIII. *The disjoint paths problem*, J. Combin. Theory Ser. B, 63 (1995), pp. 65–110.
[29] N. Robertson and P. D. Seymour, *Graph minors.* XXI. *Graphs with unique linkages*, J. Combin. Theory Ser. B, submitted.
[30] N. Robertson and P. D. Seymour, *Graph minors.* XXII. *Irrelevant vertices in linkage problems*, J. Combin. Theory Ser. B, submitted.
[31] N. Robertson and P. D. Seymour, *An outline of a disjoint paths algorithm*, in Paths, Flows, and VLSI-Layout, B. Korte, L. Lovász, H. J. Prömel, and A. Schrijver, eds., Springer-Verlag, Berlin, 1990, pp. 267–292.
[32] P. D. Seymour, *A bound on the excluded minors for a surface*, submitted.
[33] S. Stahl and L. W. Beineke, *Blocks and the nonorientable genus of graphs*, J. Graph Theory, 1 (1977), pp. 75–78.

[34] C. Thomassen, *The graph genus problem is NP-complete*, J. Algorithms, 10 (1989), pp. 568–576.

[35] C. Thomassen, *Triangulating a surface with a prescribed graph*, J. Combin. Theory Ser. B, 57 (1993), pp. 196–206.

[36] S. G. Williamson, *Embedding graphs in the plane — algorithmic aspects*, Ann. Discrete Math., 6 (1980), pp. 349–384.

[37] S. G. Williamson, *Depth-first search and Kuratowski subgraphs*, J. Assoc. Comput. Mach., 31 (1984), pp. 681–693.

SIAM J. DISCRETE MATH.
Vol. 12, No. 1, pp. 27–34

NONSYSTEMATIC PERFECT CODES*

KEVIN T. PHELPS† AND MIKE LEVAN‡

Abstract. We investigate nonsystematic perfect binary codes of length n, establishing that nonsystematic perfect codes of length n exist for all admissible $n \geq 15$. This improves on the result of Avgustinovich and Solov'eva [*Proc. 5th Internat. Workshop Algebraic and Combinatorial Coding Theory*, Cosopol, Bulgaria, 1996, pp. 15–19] who have established the existence of such codes for $n \geq 255$. We also provide the results of a computer investigation of nonsystematic perfect codes of length 15.

Key words. perfect codes

AMS subject classification. 94B25

PII. S0895480196312206

1. Introduction. A perfect single error correcting binary code has length $n = 2^r - 1$, minimum distance 3, and 2^{n-r} code words. For brevity, we will refer to such a code as simply, a perfect code (of length n). The linear perfect code, known as the Hamming code, is unique, but there are innumerable nonlinear perfect codes when $n \geq 15$. A perfect code of length $n = 2^r - 1$ is said to be systematic if there are $n - r$ coordinates such that no two code words are identical when restricted to these coordinates; otherwise it is said to be nonsystematic. Obviously, the Hamming code is systematic, but a long-standing open problem was whether there exist nonlinear perfect codes that are nonsystematic. Solutions to this problem have implications to a number of other areas including resilient functions (cf. [5]).

Recently, Avgustinovich and Solov'eva [1], [2] established the existence of nonsystematic perfect codes of length $n = 2^r - 1$, for all $r \geq 8$. Since perfect codes of length $n = 3$ and 7 are unique, this leaves the question open for perfect codes of length $n = 15,\ 31,\ 63,\ 127$. In working on a different problem as part of his dissertation [7], LeVan had generated many nonequivalent perfect codes of length 15. Subsequent examination of these codes revealed that a number of them were nonsystematic, thus settling the case $n = 15$. In this paper, we present arguments which improve on the result of Avgustinovich and Solov'eva [1], [2] establishing the existence of nonsystematic perfect codes of length $n = 2^r - 1$, for all $r \geq 6$, in particular for $n = 63$ and 127. The proofs, besides yielding better results, are also an improvement in that they are in some sense "simpler." We use a combination of argument and computation to establish the existence of a nonsystematic perfect code of length 31. Thus the question of the existence of nonsystematic perfect codes is completely settled.

Finally, we also investigate certain other related properties on nonsystematic perfect codes of length $n = 15$.

2. Preliminary results. There is a natural and well-known correspondence between binary vectors of length n and subsets of the set $\{1, 2, \ldots, n\}$. Formally,

*Received by the editors November 18, 1996; accepted for publication July 23, 1997; published electronically January 29, 1999.

http://www.siam.org/journals/sidma/12-1/31220.html

†120 Math Annex, Department of Discrete and Statistical Sciences, Auburn University, Auburn, AL 36849-5307 (phelpkt@mail.auburn.edu).

‡Brown Science Center 117, Department of Mathematics, Transylvania University, Lexington, KY 40508-1797 (mlevan@mail.transy.edu).

the support of a code word x, denoted by $supp(x)$, is just the set of coordinates of x which is equal to 1 (i.e., $supp(x) \subseteq \{1, 2, \ldots, n\}$). For any binary vectors, x and y, the distance between x and y, denoted by $d(x, y)$, is the number of coordinates in which they disagree. The set of coordinates in which they disagree is just $supp(x + y)$. We emphasize that in this paper, we are always working in the vector space over $GF(2)$.

In a perfect code C of length n, we are interested in the positions in which two code words disagree, especially when they are distance 3 apart. Formally, we define the set of triples $ST(C)$ of a perfect code C as

$$ST(C) = \{supp(x + y) | x, y \in C \text{ and } d(x, y) = 3\}.$$

In other words, a triple $\{i, j, k\}$ is in $ST(C)$ iff there exists a pair of code words x, y in C such that $\{i, j, k\} = supp(x + y)$. For the Hamming code of length $n = 2^r - 1$, (denoted by H_r), the set of triples $ST(H_r)$ is just the words of weight 3 in the code (which form a Steiner triple system on $2^r - 1$ points). In what follows, it is important to remember that a Steiner triple system is a set of three subsets or triples of an n-set such that every pair of points is covered exactly once.

Recall that a 3-uniform hypergraph is just a pair (V, E) where V is an n-set (called vertices) and E is a set of three element subsets of V (called edges). The set of triples $ST(C)$ of a perfect code C can be thought of as a 3-uniform hypergraph with $V = \{1, 2, \ldots, n\}$. Recall further that an r-subset of vertices of a 3-uniform hypergraph is said to be *stable* or *independent* if it does not contain any edge of the hypergraph. The *stability number* of a hypergraph is just the size of the largest stable set. Clearly, if a perfect code C is to be systematic on $n - r$ coordinates, then the complementary r coordinates must be a stable set in $ST(C)$; otherwise, there are two code words which disagree only in three of these r coordinates and agree everywhere else.

The following is an immediate observation.

LEMMA 1. *For a perfect code of length $n = 2^r - 1$, if the set of triples $ST(C)$ has stability number less than r, then the code is nonsystematic.*

Turán's number $T(n, r, 3)$ is just the minimum number of edges in a 3-uniform hypergraph on n points with stability number less than r (see de Caen [3]). Computing this number is a difficult problem. Avgustinovich and Solov'eva [1], [2] avoided this difficulty by constructing perfect codes C of length $n = 2^r - 1, r \geq 8$ such that the set of triples $ST(C)$ is the complete 3-uniform hypergraph (i.e., $ST(C)$ contains all three subsets of $\{1, 2, \ldots, n\}$). Obviously, such hypergraphs have stability number 2 and thus these codes are nonsystematic. In our investigation of nonsystematic perfect codes of length 15, we examine a similar problem to Turan's.

The most intuitive and basic approach to constructing (nonlinear) perfect codes involves starting with the Hamming code, H_r, and *switching* out one specially selected set of code words $S \subset H_r$ for another set of words S'. The resulting code,

$$C = (H_r \setminus S) \cup S',$$

would still be perfect as long as $|S| = |S'|$ and the minimum distance between code words in C is still 3. The difficult part is proving that one can select S and S' so that they will have the necessary properties. This general approach has a long history but we will be concerned with only more recent and relevant developments.

For the purposes of this paper, the set S can be formed by selecting cosets of linear subcodes of the Hamming code H_r. These linear subcodes are generated by all the words of weight 3 in H_r having a 1 in the same (ith) coordinate. To make things simpler, we require the chosen cosets to be mutually disjoint. There are alternate definitions of this subcode which provide different insights into its structure and properties and these insights were developed in a series of papers (see [8], [4], [6], [1], [2]).

Formally, we define a linear subcode T_i of the Hamming code H_r, for each coordinate i, as the subcode spanned by the words of weight 3 in H_r having a 1 in the (i th) coordinate. The dimension of T_i is clearly $(n-1)/2$.

We then select a subset of coordinates, $I \subseteq \{1, \dots, n\}$, and corresponding coset representatives $x_i \in H_r$, letting

$$S = \bigcup_{i \in I} T_i + x_i,$$

where

$$T_i + x_i \bigcap T_j + x_j = \emptyset, \text{ for } i \neq j.$$

Then we have

$$S' = \bigcup_{i \in I} T_i + x_i + e_i,$$

where e_i has a 1 only in the i th coordinate and zeros elsewhere. The resulting code will be perfect (see [4], [8], or [6]).

Let us consider how this switching would effect the set of triples $ST(C)$ of the perfect code C (see Fig. 2.1).

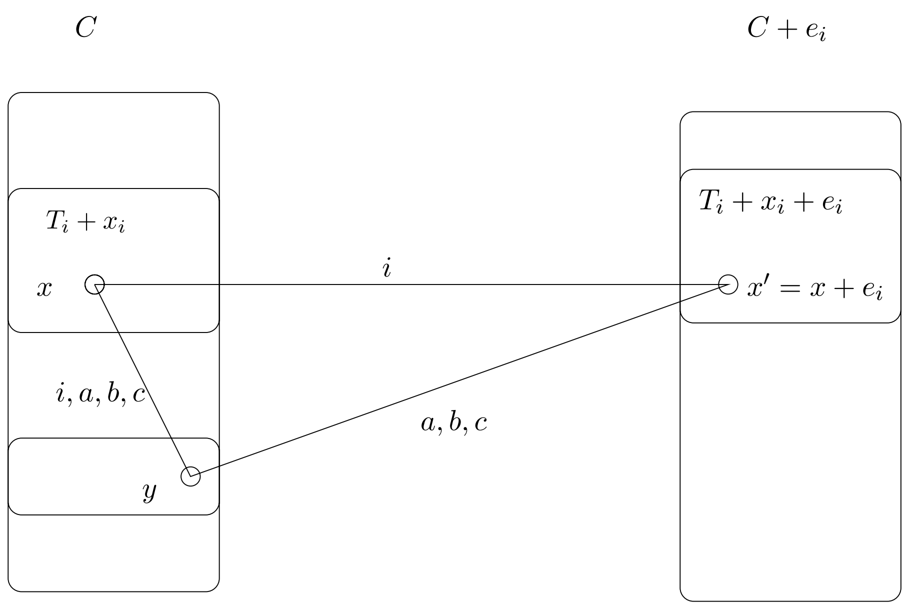


FIG. 2.1.

Consider a code word x. Let $x \in T_i + x_i$ and $x' = x + e_i$, then $x' \in T_i + x_i + e_i$. For every $y \in C$ such that $d(x, y) = 4$ and $supp(x + y) = \{i, a, b, c\}$, we have $d(x', y) = 3$ and $supp(x' + y) = \{a, b, c\}$. Since T_i is generated by the words of weight 3 in H_r

containing i, it is easy to see that $y \notin T_i + x_i$ and thus switching $T_i + x_i$ and $T_i + x_i + e_i$ will result in the triples $\{a, b, c\}$ being added to the set of triples. Of course, some code words which had been distance 3 from some $x \in T_i + x_i$ will now be distance 4 from x'. Avgustinovich and Solov'eva [1], [2] argue that if the cosets are chosen so that the distance between them is at least 5, then no triple from the set of triples of the code will be lost by any switch. Furthermore, they argue that choosing n such cosets, i.e., $|I| = n$, will ensure that the set of triples of the resulting code will be the complete uniform 3-hypergraph on n vertices. This last observation follows from the fact that because the code is perfect, every triple $\{a, b, c\}$ is either the support of a code word (and thus in $ST(C)$) or is distance 1 from a code word x of weight 4 with $supp(x) = \{i, a, b, c\}$ for some i and thus will be in $ST(C)$ after all the switches.

3. Nonsystematic codes of length n. In this section we present an argument which is a little stronger and somewhat simpler than that of Avgustinovich and Solov'eva [1], [2]. This enables us to establish that nonsystematic perfect codes exist for all $n = 2^r - 1$, $r \geq 6$. The key argument can be viewed as an extension of the proof of Lemma 5 of Phelps and LeVan [6].

Again, let T_i denote the linear subcode of the Hamming code H_r of length $n = 2^r - 1$ generated by the words of weight 3 in H_r having a 1 in the ith component. Since the dimension of T_i is $(n-1)/2$, we have that the number of cosets of T_i in H_r is $2^{2^{r-1}-r}$. The following results from [6] are crucial to our argument (and that of [1], [2]).

LEMMA 2 (see [6, Theorem 2]). *For any i, j, $i \neq j$, the intersection of linear subcodes T_i and T_j has dimension 2^{r-2}, or equivalently,*

$$|T_i \cap T_j| = 2^{2^{r-2}}.$$

COROLLARY 1 (see [6]). *Each coset $T_i + x_i$ intersects with at most $2^{2^{r-2}-1}$ cosets of T_j.*

Proof. Given cosets $T_i + x_i$ and $T_j + x_j$, if z is in the intersection of these cosets, then

$$T_i + x_i = T_i + z \text{ and } T_j + x_j = T_j + z.$$

But

$$(T_i + z) \cap (T_j + z) = (T_i \cap T_j) + z.$$

Therefore, any coset of T_j is either disjoint from $T_i + x_i$ or intersects it in $2^{2^{r-2}}$ words. Since the cosets of T_j are disjoint and $T_i + x_i$ has $2^{2^{r-1}-1}$ code words, the result follows. ∎

LEMMA 3. *In the Hamming code H_r of length $n = 2^r - 1$, $r \geq 6$ there exist n mutually disjoint cosets $T_i + x_i$, $i = 1, 2, \ldots, n$, one for each coordinate position.*

Proof. The argument is the same as in Lemma 5 of [6]. Since at most $2^{2^{r-2}-1}$ of the $2^{2^{r-1}-r}$ cosets of T_j intersect a given coset $T_i + x_i$, we can find m disjoint cosets (and thus make m switches) as long as

$$m 2^{2^{r-2}-1} < 2^{2^{r-1}-r}$$

or

$$log_2 m < 2^{r-2} - r + 1.$$

This inequality holds for $m = n$ when $r \geq 6$. □

Avgustinovich and Solov'eva [1] argue that one can find a set of disjoint cosets $T_i + x_i$, one for each coordinate, which, in addition, are at least distance 5 apart from one another. This requirement is not necessary as the following result demonstrates.

Theorem 1. *Given the Hamming code H_r of length $n = 2^r - 1$, $r \geq 5$, if one can find n disjoint cosets $T_i + x_i$, one for each coordinate i, $i = 1, 2, \ldots, n$, then there exists a nonsystematic perfect code of length n.*

Proof. Given the Hamming code H_r and the n disjoint cosets $T_i + x_i$ one removes these cosets and replaces them with the cosets $T_i + x_i + e_i$. Let C denote the perfect code which results from this *switch*. We claim that the set of triples of C, $ST(C)$, will contain every triple and thus C will be nonsystematic if $r \geq 5$.

Since the Hamming code is perfect, every binary word of weight 3, (i.e., every triple $\{a, b, c\}$), is either a code word (and thus in the set of triples of the Hamming code) or distance 1 from a code word z, of weight 4. Let $supp(z) = \{i, a, b, c\}$ and assume that $T_i + x_i + e_i$ replaced the coset $T_i + x_i$ in H_r. As we noted above, for each $x' = x + e_i$ of C, $x' \in T_i + x_i + e_i$ there is a unique $y = x + z$, $y \in T_i + x_i + z$ such that $supp(x' + y) = \{a, b, c\}$. This triple, $\{a, b, c\}$, will remain in the set of triples of C as long as there is at least one such pair of code words x', y. Some of these code words y may be in other chosen cosets $T_j + x_j$ and thus would not be in C. But this will not matter as long as some of these code words $y \in T_i + x_i + z$ are left. Again, assuming $n = 2^r - 1$ and one wants to pick $m \leq n$ disjoint cosets to switch out, then using the fact that any switch replaces at most $2^{2^{r-2}}$ code words of the coset $T_i + x_i + z$, there must be some code words remaining in C as long as

$$m 2^{2^{r-2}} < 2^{2^{r-1}-1}.$$

Simplifying this gives

$$log_2 m < 2^{r-2} - 1.$$

This inequality holds for $m = n = 2^r - 1$ and $r \geq 5$. Thus in this case, we only need the n chosen cosets, one for each coordinate position, to be disjoint for the resulting code C to be nonsystematic! □

Since we can always find such a set of disjoint cosets when $r \geq 6$ by the previous lemma, we have the following corollary.

Corollary 2. *There exist nonsystematic perfect codes of length $n = 2^r - 1$ for all $r \geq 6$.*

We now turn our attention to the question of nonsystematic perfect codes of lengths 15 and 31 as well as some related questions.

4. Nonsystematic codes of length 15, 31. In our computer investigation of nonsystematic perfect codes of length 15, we examined the previously generated perfect codes of length 15 and found a number which were nonsystematic. However, none of these codes had the complete 3-uniform hypergraph as its set of triples, $ST(C)$.

We developed two approaches to the problem. The first approach investigated the minimum number of switches necessary to convert the Hamming code H_4 into a nonsystematic code. We considered all codes formed from the Hamming code by seven or fewer switches. Phelps and LeVan [6] showed that the first four independent switches that can be made will yield an equivalent code. From here, there are 17 choices one could make for the fifth switch, which leads to over 250 choices one could make for the sixth switch, which also leads to over 5000 choices one could make for the seventh switch. None of the codes resulting from six switches or less yielded a nonsystematic code. However, this approach found 144 different nonsystematic perfect codes of length 15, constructed with $m = 7$ switches. It was also determined that for all of the 144 different nonsystematic codes constructed, $|ST(C)| = 427$. So $ST(C)$ does not form the complete 3-uniform hypergraph. These codes also appear to be equivalent as the invariants we usually compute for perfect codes could not distinguish between them.

The second approach considered was to randomly make a switch from the code C to form a new code C' iff $|ST(C)| < |ST(C')|$. This approach showed that it is possible to construct a code, C, which does indeed have $|ST(C)| = 455$, and thus $ST(C)$ will form the complete 3-uniform hypergraph, but it takes at least eight switches.

Here is a listing of the generator matrix for H_4, as well as the switches and coset representatives one could use in order to construct a nonsystematic code. We have also listed the size of $ST(C)$ after every switch.

$$H_4 = \left\{ \begin{array}{llll} 010100000000101, & 000000110000110, & 111100000000000, & 000000000000111, \\ 000110001000111, & 000001010000101, & 000000000011110, & 000000001111000, \\ 000011110000000, & 001100000000110, & 000000000101101 & \end{array} \right\}$$

Switch	Coset representative	x_i	$\|ST(C)\|$	Systematic
0	–	–	35	yes
1	$T_{12} + e_{12} + x_1$	110010100000000	63	yes
2	$T_6 + e_6 + x_2$	000010000000101	107	yes
3	$T_3 + e_3 + x_3$	110111000101101	167	yes
4	$T_2 + e_2 + x_4$	001001010101111	243	yes
5	$T_{13} + e_{13} + x_5$	001000101101100	311	yes
6	$T_7 + e_7 + x_6$	101110100010001	375	yes
7	$T_8 + e_8 + x_7$	000001011000101	427	no
8	$T_9 + e_9 + x_8$	001100000111000	455	no

This settles the case for $n = 15$. We now turn our attention to the case when $n = 31$.

To construct a nonsystematic perfect code of length 31, we just have to find 31 disjoint cosets of T_i to switch out. The arguments of the previous section only prove that one can find 15 disjoint cosets $T_i + x_i$, but by a computer search we found all the switches that are needed to construct the desired code. We generated 31 disjoint cosets, $T_i + x_i$, one for each coordinate $i = 0, \ldots, 30$. Making these switches would give us a code whose set of triples, $ST(C)$, was again the complete uniform 3-hypergraph on 31 vertices, which also says the code is nonsystematic.

We start with a cyclic representation of the Hamming code of length 31. The supports for the words of weight 3 are all cyclic shifts of the following:

$$\{0,1,12\}, \{0,2,24\}, \{0,3,8\}, \{0,4,17\}, \{0,6,16\}.$$

The coset T_0 is generated by the words of weight 3 containing 0. The supports for these words are the following triples:

$$\begin{array}{lllll} \{0,1,12\}, & \{0,2,24\}, & \{0,3,8\}, & \{0,4,17\}, & \{0,6,16\}, \\ \{0,7,9\}, & \{0,10,25\}, & \{0,11,30\}, & \{0,13,27\}, & \{0,14,18\}, \\ \{0,15,21\}, & \{0,19,20\}, & \{0,22,29\}, & \{0,23,26\}, & \{0,5,28\}. \end{array}$$

The generators for the coset T_i then are just the ith cyclic shift of these words. The coset representatives x_i for each i are the following:

i	x_i	i	x_i
0	0000000000001000000000000010001	16	0010000000010111000000000000000
1	0000000001000000000000000000110	17	0001000000001101100000000100000
2	0100000000000010000000000000100	18	0000100000000101110000000000000
3	1000000000010000000000000000001	19	0000010000000011011000000001000
4	0001000000000000100000000000001	20	0100000000000001000100000000000
5	0110000000000100000000000000000	21	0000000000000000011000000000010
6	0111010000000010001000000000000	22	0001000000000000010001000000000
7	0001100000000001000000000000000	23	0000100001000000000011000000000
8	0001110100000000100010000000000	24	0000010000000000000100010000000
9	0000011000000000010000000000000	25	0000000000010000000010111000000
10	0000011101000000001000100000000	26	0001000100001000000000101000000
11	0000000110000000000100000000000	27	0000000000001000000000101110000
12	0000000111010000000010001000000	28	0000000001000010000000000110000
13	0000000001100000000001000000000	29	0000001000100001000000000101000
14	0000000001110100000000100010000	30	0000000100010000000000000111010.
15	0100000000110110000000010000000		

We claim that these cosets are mutually disjoint and thus by Theorem 1, replacing each coset, $T_i + x_i$ by $T_i + x_i + e_i$, will produce a nonsystematic perfect code of length 31.

5. Conclusions and questions. In summary, we have established the following.

THEOREM 2. *There exists a nonsystematic perfect binary single error correcting code of length* $n = 2^r - 1$ *iff* $r \geq 4$.

From the previous arguments and computations we can extend another result of Avgustinovich and Solov'eva [1], [2].

THEOREM 3. *There exists a nonlinear perfect binary single error correcting code of length* $n = 2^r - 1$*, whose set of triples is the complete* 3*-uniform hypergraph of order* n *iff* $r \geq 4$.

There are several additional questions regarding nonsystematic perfect codes which are of interest. The one which intrigues us is what is the fewest number of switches needed to change the Hamming code into a nonsystematic perfect code? For instance, we believe that one can form a nonsystematic code of length 31 by choosing far fewer switches than the 31 switches we used. Related to this is Turan's problem and how small can the set of triples, $ST(C)$, of the nonsystematic perfect code C be?

Finally, the fact the $ST(C)$ has stability number less than r (for $n = 2^r - 1$) is sufficient for a code to be nonsystematic. Is it necessary? This seems unlikely.

REFERENCES

[1] S. V. AVGUSTINOVICH AND F. I. SOLOV'EVA, *Existence of nonsystematic perfect binary codes*, in Proc. 5th Internat. Workshop Algebraic and Combinatorial Coding Theory, Cosopol, Bulgaria, 1996, pp. 15–19.

[2] S. V. AVGUSTINOVICH AND F. I. SOLOV'EVA, *On nonsystematic perfect binary codes*, Problemy Peredachi Informatsii, 32 (1996), pp. 47–50 (in Russian).

[3] D. DE CAEN, *The current status of Turán's problem on hypergraphs*, in Extremal Problems for Finite Sets, Visegrad, Hungary, 1991, Bolyai Soc. Math. Stud. 3, János Bolyai Math. Soc., Budapest, 1994, pp. 187–197.

[4] T. ETZION AND A. VARDY, *Perfect codes: Constructions, properties and enumeration*, IEEE Trans. Inform. Theory, 40 (1994), pp. 754–763.

[5] T. Etzion and A. Vardy, *On perfect codes and tilings: Problems and solutions*, SIAM J. Discrete Math., 11 (1998), pp. 205–223.
[6] K. T. Phelps and M. LeVan, *Kernals of nonlinear Hamming codes*, Des. Codes Cryptogr., 6 (1995), pp. 247–257.
[7] J. M. LeVan, *Designs and Codes*, Dissertation, Auburn University, Auburn, AL, June 1995.
[8] F. I. Solov'eva, *Factorization of code-forming DNFs*, Metody Diskret. Analiz., 47 (1988), pp. 66–88 (in Russian).

SIAM J. DISCRETE MATH.
Vol. 12, No. 1, pp. 35–47

WELL-ORDERED STEINER TRIPLE SYSTEMS AND 1-PERFECT PARTITIONS OF THE N-CUBE*

JOSEP RIFÀ†

Abstract. Binary 1-perfect codes which give rise to partitions of the n-cube are presented. The 1-perfect partitions are characterized as homomorphic images of simple algebraic structures on $\boldsymbol{F}^n$ and are constructed starting from a particular case of a structure defined in $\boldsymbol{F}^n$.

A special property (so-called *well-ordering*) of $STS(n)$ is given in such a way that for this kind of STS it is possible to define the algebraic structure we need in $\boldsymbol{F}^n$ and to construct 1-perfect partitions of the n-cube.

These 1-perfect partitions give us a kind of 1-perfect code for which it is easy to do the coding and decoding. Furthermore, there exists a syndrome which allows us to perform error correction. We present systematic codes of length $n = 15$ and we give examples of how to do the coding, decoding, and error correction.

Key words. 1-perfect binary codes, 1-perfect partitions, Steiner triple systems, Sloops, distance-compatible action

AMS subject classifications. 94B25, 05B30, 68R05

PII. S0895480197330722

1. Introduction.

Let $\boldsymbol{F}$ be the binary finite field $GF(2)$ and consider the n-cube $\boldsymbol{F}^n$.

A binary code C of length n is a subset of $\boldsymbol{F}^n$. If this subset is a linear subspace of $\boldsymbol{F}^n$, then C will be a linear code. In any case we will call the vectors in C codewords.

The concept of Hamming distance between two vectors $v, w \in \boldsymbol{F}^n$ is defined as the number of coordinates in which they differ. A binary code is a 1-perfect code if all the vectors in $\boldsymbol{F}^n$ are either in C or at distance one from exactly one codeword of C.

A binary 1-perfect code has length $n = 2^m - 1$, and the linear 1-perfect codes are unique up to isomorphism (see [4]). The characterization of binary nonlinear 1-perfect codes is not complete. Nonlinear 1-perfect codes were first constructed by Vasil'ev, and other constructions have been presented subsequently by Mollard, Phelps, Solov'eva, Bauer, and more recently by Etzion and Vardy (the reader can see a review of all these constructions in [2]).

Two 1-perfect codes are isomorphic if there exists a permutation of the coordinates such that the codewords in the first code are converted to the codewords in the second code.

Two 1-perfect codes are equivalent if there exists a translation such that the codewords in the first code are converted to the codewords in the second code or isomorphic to them (see [7]).

In this paper a construction of 1-perfect partitions of $\boldsymbol{F}^n$ is proposed, that is, partitions of the n-cube in 1-perfect codes. The construction is based on Theorems 3.1 and 3.2, which we present in section 3. In particular, within the various possibilities offered by these theorems, we have opted to use the Steiner loop (Sloop) structure associated with the well-ordered Steiner triple system (STS).

*Received by the editors December 1, 1997; accepted for publication March 10, 1998; published electronically January 29, 1999. This research was partially supported by Spanish grant TEL97-0663.
http://www.siam.org/journals/sidma/12-1/33072.html

†Department of Computer Sciences, Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain (jrifa@ccd.uab.es).

In section 2, we present several general characteristics of quasi groups, Sloops, STSs, and 1-perfect codes.

In section 3 (the main section), we look at Theorems 3.1 and 3.2, which allow the algebraic construction of 1-perfect partitions and, therefore, of 1-perfect codes. Using these theorems and the Sloop structure given by possible STSs, we see in Theorem 3.5 that, for a specific type of STS, we can ensure the construction of 1-perfect partitions of the n-cube.

In section 4, we analyze the well-ordered STSs, and in section 5 we see an example of how to handle coding, decoding, and error correction using the 1-perfect codes constructed.

Finally, we present our conclusions in section 6, along with possibilities for future research on this topic.

2. Sloops, STSs, and 1-perfect codes.

DEFINITION 2.1. *Let A, F be two sets. We say that A acts on F by means of $\cdot$ if there exists a map*

$$\begin{array}{ccc} F \times A & \longrightarrow & F, \\ (f, a) & \longrightarrow & f \cdot a. \end{array}$$

DEFINITION 2.2. *Assume A acts on F by means of $\cdot$ and also on G by means of $*$. An A-homomorphism $h : F \longrightarrow G$ is a map compatible with the action of A on F and on G, that is, a map such that for all $a \in A$, $f \in F$ it holds that $f \cdot a = h(f) * a$.*

We are interested in algebraic structures defined on the n-cube $\boldsymbol{F}^n$, and also in the Hamming distance defined between vectors in $\boldsymbol{F}^n$.

Let $\boldsymbol{A}_n$ be the set $\{e_0, e_1, \ldots, e_n\}$, where $e_0 \in F^n$ is the zero vector and e_i $(i = 1, 2, \ldots, n)$ are the basis vectors in $\boldsymbol{F}^n$ having a one in the ith coordinate and zeroes elsewhere.

DEFINITION 2.3. *A distance-compatible (Hamming distance) action of the set $\boldsymbol{A}_n$ on $\boldsymbol{F}^n$ is a map*

$$\begin{array}{ccc} \boldsymbol{F}^n \times \boldsymbol{A}_n & \rightarrow & \boldsymbol{F}^n, \\ (v, e_i) & \rightarrow & v \cdot e_i, \end{array}$$

such that

- *for all $v \in \boldsymbol{F}^n$ there is a permutation π_v of n coordinates such that $v \cdot e_i = v + e_{\pi_v(i)}$;*
- *for all $e_i \in \boldsymbol{A}_n$ the induced map $v \rightarrow v \cdot e_i$ is one-to-one.*

For instance, the translation $(v, e_i) \rightarrow v + e_i$ is a distance-compatible action of $\boldsymbol{A}_n$ on the n-cube.

The following proposition shows us three properties of distance-compatible actions of $\boldsymbol{A}_n$ on $\boldsymbol{F}^n$ that we will give without proof, because they proceed directly from the definition.

PROPOSITION 2.4.

1. *For all $v \in \boldsymbol{F}^n$ we have $v \cdot e_i = v \cdot e_j$ if and only if $i = j$.*
2. *For all e_i, $d(v \cdot e_i, v) = 1$.*
3. *The set $\{a \cdot e_i | i = 1..n\}$ is the set of all the vectors in $\boldsymbol{F}^n$ at distance one from a given $a \in \boldsymbol{F}^n$.*

One of the simplest algebraic structures is that of a quasi group, which we will use in this paper. Readers interested in quasi groups and related structures can find more information in [6].

DEFINITION 2.5. *Let A be a finite set. An algebraic structure of a quasi group consists of A and a binary operation on A defined by the function*

$$* : A \times A \longrightarrow A$$

such that $x * y = x * z$ and $y * x = z * x$ only if $y = z$ for all x, y, $z \in A$.

DEFINITION 2.6. *A quasi group $(A, *)$ is called a Sloop if*

- *there exists $0 \in A$ such that $0 * a = a * 0 = a$ for all $a \in A$;*
- *the operation is totally symmetric, that is, any relation $a * b = c$ implies any other relation obtained by permuting a, b, and c.*

DEFINITION 2.7. *A Steiner triple system $STS(n)$ is a pair (A, B), where A is a finite set of n elements and B is a collection of 3-subsets of A, which we will call blocks, such that every two different elements $x, y \in A$ are contained in exactly one block of B.*

- It is easy to see that starting from a Sloop A, we can define an STS on the set $A^* = A - \{0\}$ by taking a set of blocks $B = \{(x, y, x * y) \,|\, \forall\, x, y \in A^*,\ x \neq y\}$.
- Conversely, starting from an $STS(n) = (A^*, B)$, we can define a Sloop on the set $A = A^* \cup \{0\} = \{0, 1, 2, \ldots, n\}$ by

$$\begin{array}{ccc} A \times A & \longrightarrow & A, \\ (a, b) & \longrightarrow & a * b, \end{array}$$

$$\begin{cases} \text{if } a \neq b & \text{then} \quad a * b = c, \quad \text{where } (a, b, c) \in B, \\ \text{if } a = b & \text{then} \quad a * b = 0, \\ \text{if } a = 0 & \text{then} \quad a * b = b, \\ \text{if } b = 0 & \text{then} \quad a * b = a \end{cases}$$

- Two STSs (A, B) and (A', B') are isomorphic if $A = A'$ and there exists a permutation of the elements in A such that the triples in B are converted to the triples in B'.
  If $\|A^*\| = 15$, there are 80 nonisomorphic triples (see [9]).
  If $\|A^*\| = 31$, there are $\approx 10^{200}$ nonisomorphic triples (see [5]).

Starting from a 1-perfect binary code $C \in \boldsymbol{F}^n$ (not necessarily linear but such that $0 \in C$), we can construct an STS by taking the supports of the codewords of weight three. Take $A^* = \{1, 2, \ldots, n\}$ as the set of coordinates, and the set of blocks as $B = \{(i, j, k)\}$, where (i, j, k) are the support of any codeword in C of weight three. We denote this set by STS_0.

Let C be a 1-perfect binary code. Let $v \in C$ be a codeword in C. The set of all $w \in C$ at distance three from v is an STS_v taking as the set of blocks B the support of all the vectors $v + w$ $(\forall\, w \in C \,|\, d(w, v) = 3)$.

Starting from a 1-perfect code C we can obtain different STSs, for instance STS_0, STS_v, etc.

An STS can be obtained from a 1-perfect code or not. In the case that the STS comes from a 1-perfect code, it can be unique or not and, moreover, if there is more than one 1-perfect code which gives the same STS, they do not need to be isomorphic nor equivalent.

Phelps (see [7]) constructs several 1-perfect codes in a combinatorial way which lead to 23 of 80 nonisomorphic STSs of length 15 (these STSs are called "perfect"). Levan (see [3]) adds 8 codes to the previous list.

In this paper we prove that starting from a well-ordered STS it is possible to construct a partition of $\boldsymbol{F}^n$ in 1-perfect codes such that the given STS is the support of the minimum-weight codewords. It will remain the same problem when the given STS is not well ordered.

3. 1-perfect partitions. In this paper, we are interested in 1-perfect codes which give rise to partitions of $\boldsymbol{F}^n$ in 1-perfect codes, rather than in 1-perfect codes alone.

We already know that, given any 1-perfect code C of length n, we can always find a partition of $\boldsymbol{F}^n$ generated by this code. For example, the trivial partition $\{C_i \mid C_i = C + e_i;\ \forall\, i = 1, \ldots, n\}$, where e_i are the different vectors of $\boldsymbol{F}^n$ of weight 1 and $e_0 = (0, 0, \ldots, 0)$, is a partition of $\boldsymbol{F}^n$ on 1-perfect codes, that is, a 1-perfect partition. The above partition is only natural when C is a linear code, that is, in those cases where $C + C = C$.

In other 1-perfect codes, another type of partition would be more natural. For example, in propelinear codes (see [8]), it would be more natural to use the partition on $\boldsymbol{F}^n$ given by $\{C_i \mid C_i = C * e_i;\ \forall\, i = 1, \ldots, n\}$ since, for these codes, $C * C = C$.

Generally, for 1-perfect codes, there does not exist an operation on $\boldsymbol{F}^n$ allowing a natural partition. There is a gap in the literature on this aspect and this paper attemps to analyze it.

We begin by assuming that we have $\boldsymbol{F}^n$ partitioned into classes, each of which is a 1-perfect code. In every class other than the class C, which includes the vector 0, we can take a vector of weight 1 as a representative and, therefore, we can consider the partition as given by $\boldsymbol{A}_n = \{e_0, e_1, e_2, \ldots, e_n\}$.

THEOREM 3.1. *Given a 1-perfect partition $\boldsymbol{A}_n$ on $\boldsymbol{F}^n$ it is possible to define a distance-compatible action of $\boldsymbol{A}_n$ on $\boldsymbol{F}^n$, such that the given partition can be considered as a quasi group which is an $\boldsymbol{A}_n$-homomorphic image of $\boldsymbol{F}^n$.*

Proof. In essence, we assume a 1-perfect partition $\boldsymbol{A}_n$ and define an operation on $\boldsymbol{A}_n$ as follows:

$$e_i * e_j = e_k, \tag{3.1}$$

where e_k represents the class containing the vector $e_i + e_j$. $\boldsymbol{A}_n$ has a quasi group structure with this operation, where e_0 is the zero element. In fact $\boldsymbol{A}_n$ has a Sloop structure.

This operation is not the only one which could be defined on $\boldsymbol{A}_n$.

Assuming that $\boldsymbol{A}_n$ has a quasi-group structure, it is important to observe whether $\boldsymbol{A}_n$ can be considered as an $\boldsymbol{A}_n$-homomorphic image of $\boldsymbol{F}^n$. For this purpose, we must have defined an operation on $\boldsymbol{F}^n$ or at least an operation between elements of $\boldsymbol{F}^n$ and $\boldsymbol{A}_n$ ($\boldsymbol{A}_n$ could be considered a subset of $\boldsymbol{F}^n$).

Given any element $c \in C$, we define $c \cdot e_i$ as the only element of class e_i at distance one from c.

Given any element $v \in \boldsymbol{F}^n$, since C is a 1-perfect code, we can always write it uniquely as $v = c \cdot e_i$, where $c \in C$. We now define an operation $\boldsymbol{F}^n \times \boldsymbol{A}_n \longrightarrow \boldsymbol{F}^n$ such that $v \cdot e_j = w$ is the only vector of the class $e_k \in \boldsymbol{A}_n$ at distance one from v, where $e_i * e_j = e_k$.

This operation $\boldsymbol{F}^n \times \boldsymbol{A}_n$ meets the conditions of Definition 2.3, so we have a distance-compatible action of $\boldsymbol{A}_n$ on $\boldsymbol{F}^n$.

Now we can define

$$\phi : \boldsymbol{F}^n \longrightarrow \boldsymbol{A}_n$$

such that $\phi(v) = e_i$ if and only if v is in class e_i.

If $\phi(v \cdot e_j) = e_k$, then $e_i * e_j = e_k$, where $\phi(v) = e_i$. Hence $\phi(v) * \phi(e_j) = e_k$ and $\phi(v \cdot e_j) = \phi(v) * \phi(e_j)$, so ϕ is an $\boldsymbol{A}_n$-homomorphism, which is the identity map on $\boldsymbol{A}_n$. ∎

Now the inverse: let us assume that we have defined a distance-compatible action of $\boldsymbol{A}_n$ on $\boldsymbol{F}^n$ and also that we have defined a quasi-group structure with a zero element on $\boldsymbol{A}_n$.

With these conditions, we will consider the following theorem.

THEOREM 3.2. *Let us assume there is an $\boldsymbol{A}_n$-homomorphism $\phi : \boldsymbol{F}^n \longrightarrow \boldsymbol{A}_n$ which is the identity map on $\boldsymbol{A}_n$.*

Then, for all $e_i \in \boldsymbol{A}_n$, the sets $H = \phi^{-1}(e_i) \subset \boldsymbol{F}^n$ are 1-perfect codes.

Proof. First, we will see that the minimum distance of H is 3.

Suppose $d(a,b) = 1$, where $a, b \in H$. For some index j, $a \cdot e_j = b$, since all $a \cdot e_j$ are different and we obtain the elements of $\boldsymbol{F}^n$ at distance one from a. Hence, $\phi(b) = \phi(a \cdot e_j) = \phi(a) * \phi(e_j) = \phi(a) * e_j$, but $\phi(a) = \phi(b)$, so $e_0 = e_j$, which contradicts the initial assumption.

Let us now assume $d(a,b) = 2$, where $a, b \in H$. There will be $e_i \neq e_j$ such that $a \cdot e_i = b \cdot e_j$. Hence $\phi(a \cdot e_i) = \phi(b \cdot e_j)$ and, since $\phi(a) = \phi(b)$, then $\phi(e_i) = \phi(e_j)$ and, therefore, $e_i = e_j$, which is impossible.

Finally, we will see that, given any element $v \in \boldsymbol{F}^n$, then either $v \in H$, or there is a unique element $w \in H$ such that $d(v, w) = 1$.

In essence, let us assume that $v \notin H$ and $\phi(H) = e_k$. Then for any index i, $\phi(v) = e_i$. Since $\forall\, j \in \boldsymbol{A}_n$, $j \neq 0$, the elements $e_i * e_j \in \boldsymbol{A}_n$ are all different, there will be a certain value for which $e_i * e_j = e_k$. Hence $\phi(v \cdot e_j) = \phi(v) * e_j = e_k$ and $w = v \cdot e_j \in H$. Moreover, $d(v, w) = 1$.

Suppose now that there is a $w' \in H$, $w' \neq w$ at distance 1 from v. This means that, for a certain s, we have $w' = v \cdot e_s$ and $\phi(w') = e_k$. Therefore, $e_i * e_s = e_k = e_i * e_j$ and $e_s = e_j$, contrary to what we assumed. □

According to this theorem, our interest lies, therefore, in defining distance-compatible actions of $\boldsymbol{A}_n$ on $\boldsymbol{F}^n$ for which $\boldsymbol{A}_n$ is a homomorphic image.

One way to do so is the following.

Fix an order in the set $\boldsymbol{A}_n - \{e_0\}$; for instance $e_1 < e_2 < e_3 \cdots < e_n$.

For $x \in \boldsymbol{F}^n$, $x = (x_1, x_2, \ldots, x_n)$ define the ordered support of x as $s_x = e_{a_1} < e_{a_2} < \cdots < e_{a_r}$, where $e_{a_i} \in s_x$ if and only if $x_{a_i} = 1$.

Given an $STS(n)$ we can define in $\boldsymbol{A}_n = \{e_0, e_1, \ldots, e_n\}$ the Sloop structure as we stated in Definition 2.7. Hence, if B is the set of blocks of the given STS we have

$$(3.2)\qquad \begin{cases} \text{if } e_i \neq e_j, & \text{then} \quad e_i * e_j = e_k, \quad \text{where } (e_i, e_j, e_k) \in B, \\ \text{if } e_i = e_j, & \text{then} \quad e_i * e_j = 0, \\ \text{if } e_i = e_0, & \text{then} \quad e_i * e_j = e_j, \\ \text{if } e_j = e_0, & \text{then} \quad e_i * e_j = e_i. \end{cases}$$

For $x \in \boldsymbol{F}^n$ define the value $\phi(x)$ of x in the following way:

$$(3.3)\qquad \begin{array}{ccc} \boldsymbol{F}^n & \stackrel{\phi}{\longrightarrow} & \boldsymbol{A}_n, \\ x & \longrightarrow & \phi(x) = ((\cdots((e_{a_1} * e_{a_2}) * e_{a_3}) * e_{a_4} * \cdots) * e_{a_r}), \end{array}$$

where $s_x = e_{a_1} < e_{a_2} < \cdots < e_{a_r}$.

Given $c_1, c_2, \ldots, c_r$, we will write $[c_1 c_2 \cdots c_r] \in \boldsymbol{A}_n$ to represent the result of the chain of operations $((\cdots((c_1 * c_2) * c_3) * c_4 * \cdots) * c_r)$.

Given $a, x, y \in \boldsymbol{A}_n$, the equation $(a * x) * y = (a * \bar{y}) * x$ always has a unique solution that can be calculated as

$$(3.4)\qquad \bar{y} = [axyxa].$$

For some STS the condition $\bar{y} = y$ is always true, for example, when we consider the first STS of the 80 possible STSs of length 15 (we will consider the list of 80 STSs to be ordered normally, as, for example, in [1]).

DEFINITION 3.3. *We will say that an STS is a well-ordered STS if it is possible to order the elements in $\boldsymbol{A}_n$ such that $\forall\, a, x, y \in \boldsymbol{A}_n$ we have $x < y$ if and only if $x < \bar{y}$, where $\bar{y} = [axyxa]$.*

LEMMA 3.4. *Let (A^*, B) be a well-ordered STS and let $\boldsymbol{A}_n$ be the Sloop defined in* (3.2).

Then there is a distance-compatible action of $\boldsymbol{A}_n$ on $\boldsymbol{F}^n$ such that the value map $\phi : \boldsymbol{F}^n \longrightarrow \boldsymbol{A}_n$ defined in (3.3) *is an $\boldsymbol{A}_n$-homomorphism.*

Proof. Let $x = (x_1, x_2, \ldots, x_n) \in \boldsymbol{F}^n$, and let $s_x = e_{a_1} < e_{a_2} \cdots < e_{a_r}$ be the ordered support of x.

Then $\phi(x) * e_i = [e_{a_1} e_{a_2} \cdots e_{a_r} e_i] = [e_{a_1} e_{a_2} \cdots e_{a_{r-1}} e_{i'} e_{a_r}] = [e_{a_1} e_{a_2} \cdots e_{a_{r-2}} e_{i''} e_{a_{r-1}} e_{a_r}] = \ldots$, where $e_i < e_{a_r}$ if and only if $e_{i'} < e_{a_r}$ and $e_{i'} < e_{a_{r-1}}$ if and only if $e_{i''} < e_{a_{r-2}}$.

The same argument brings us finally to an index j such that $\phi(x) * e_i = [e_{a_1} \cdots e_{a_{s-1}} e_j e_{a_s} \cdots e_{a_r}]$, where $e_{a_{s-1}} \leq e_j < e_{a_s}$.

Now we define $\pi_x(e_i) = e_j$. π_x is a permutation of $\{e_i \,|\, i = 1..n\}$ that allows us to define, for all $x \in \boldsymbol{F}^n$,

$$x \cdot e_i = x + e_j = x + \pi_x(e_i)$$

so that $\boldsymbol{A}_n$ acts on $\boldsymbol{F}^n$ and this is a distance-compatible action.

Furthermore, with the given definition, $\phi(x) * \phi(e_i) = \phi(x + e_j) = \phi(x \cdot e_i)$, so ϕ is an $\boldsymbol{A}_n$-homomorphism. ∎

As a consequence of Theorem 3.2 and Lemma 3.4, we can establish the following theorem which proves that the well-ordered property of STSs is of interest because it allows us to start from an $STS(n)$ and efficiently determine when there is a 1-perfect partition associated with it.

THEOREM 3.5. *Let (A^*, B) be a well-ordered $STS(n)$ and let $\boldsymbol{A}_n$ be the Sloop defined in* (3.2).

Then there is a distance-compatible action of $\boldsymbol{A}_n$ on $\boldsymbol{F}^n$ such that the value map $\phi : \boldsymbol{F}^n \longrightarrow \boldsymbol{A}_n$ gives us a partition of $\boldsymbol{F}^n$ into 1-perfect codes $H = \phi^{-1}(e_i)$ for all $e_i \in \boldsymbol{A}_n$.

Starting from a well-ordered STS not only can we assure that $\boldsymbol{A}_n$ acts in a distance-compatible way on $\boldsymbol{F}^n$ but we can extend the action to all the elements in $\boldsymbol{F}^n$ as we can see in the following proposition.

PROPOSITION 3.6. *Let (A^*, B) be a well-ordered STS and let $\boldsymbol{A}_n$ be the Sloop defined in* (3.2).

Then we can extend the action of $\boldsymbol{A}_n \subset \boldsymbol{F}^n$ on $\boldsymbol{F}^n$ to an action of $\boldsymbol{F}^n$ on $\boldsymbol{F}^n$.

Proof. Given $x, y \in \boldsymbol{F}^n$ with ordered supports $s_x = e_{a_1} < e_{a_2} \cdots < e_{a_r}$ and $s_y = e_{b_1} < e_{b_2} < \cdots < e_{b_s}$, respectively, we define $x \cdot y$ by using Lemma 3.4:

$$x \cdot y = (\cdots((x \cdot e_{b_1}) \cdot e_{b_2}) \cdots) \cdot e_{b_s}.$$

It is now easy to see that the previous operation is well defined, that is, $x \cdot y$ has a unique value, so we have an action of $\boldsymbol{F}^n$ on $\boldsymbol{F}^n$. ∎

Remark. Proposition 3.6 shows us that the well-ordered condition is stronger than needed to assure the construction of 1-perfect partitions starting from an STS (see Theorem 3.2). We will see in the following section that in the specific case $n = 15$ we can construct 1-perfect partitions starting in 16 $STS(15)$s, but this result does not close the problem of finding all possible STSs which allow the construction of 1-perfect partitions.

A problem we leave open is the construction of distance-compatible actions of $\boldsymbol{A}_n \subset \boldsymbol{F}^n$ on $\boldsymbol{F}^n$ that cannot be extended to actions of $\boldsymbol{F}^n$ on $\boldsymbol{F}^n$.

4. Well-ordered STSs. We will now consider the STSs which have the well-ordered property.

In general, if the equality $(a * x) * y = (a * y) * x$ does not hold, we can calculate $\bar{y} = [axyxa]$ (see (3.4)) such that $(a*x)*y = (a*\bar{y})*x$ and, if the STS is well ordered, we obtain an element $\bar{y}$ that has the same order relationship with x that y has with x.

Whenever we vary $a \in \boldsymbol{A}_n$ in (3.4), we obtain n elements, not necessarily different, that are greater than x if $y > x$, or less than x if $y < x$. For all $x \neq y$, we will use q_{xy} to designate the set of different elements obtained:

$$q_{xy} = \{\bar{y} \in \boldsymbol{A}_n | \bar{y} = [axyxa] | a \in \boldsymbol{A}_n\}$$

The vector $(q_1, q_2, \ldots, q_n)$, where q_i is the quantity of pairs (x, y) for which $|q_{xy}| = i$, will be denoted the characteristic vector of the $STS(n)$ and, when $n = 15$, it is a complete invariant for STSs which allows us to distinguish completely nonisomorphic $STS(15)$s.

In the appendix, we have listed the 80 vectors which characterize the nonisomorphic $STS(15)$s. We have suppressed the coordinates $q_8, q_9, q_{10}, q_{11}, q_{12}, q_{13}, q_{14}, q_{15}$ in each vector since their value is always zero. Moreover, we have added to each vector a coordinate q_{16} which allows us to decide which $STS(15)$s are well ordered, as we will see in Proposition 4.1.

There are other invariants which make it possible to distinguish between nonisomorphic $STS(15)$s, for example the *cycle structure* (see [5]), the *train* (see [5]), and the *fragments* (see [3]). We will use the invariant we propose, since it allows us to link the STS structure with the construction of perfect codes, as we will see later on.

All of the elements in q_{xy}'s have the same order relationship with x that y has with x.

Let us assume that for certain $y, y', y'' \in \boldsymbol{A}_n^*$ we have some elements $\alpha, \beta, \gamma \in A^*$ such that

$$\begin{aligned} \alpha, \beta &\in q_{\gamma y}, \\ \alpha, \gamma &\in q_{\beta y'}, \\ \gamma, \beta &\in q_{\alpha y''}. \end{aligned} \tag{4.1}$$

We will use q_{16} to denote the quantity of triples α, β, γ that satisfy (4.1).

PROPOSITION 4.1. *The component q_{16} in the characteristic vector of a well-ordered $STS(n)$ is zero (see the appendix to see the values of the q_{16} for all the $STSs$ of length* 15*).*

Proof. In essence, if $q_{16} \neq 0$, then there is a α, β, γ triple that fulfills (4.1). Nevertheless, this is absurd since, if $\alpha > \beta > \gamma$, the second equation fails; if $\alpha > \gamma > \beta$, the first equation fails, etc. For any assumption, one of the three equations in (4.1) always fails. □

Proposition 4.1 limits the number of $STS(15)$s for which it is possible to define a well-ordering that allows to obtain perfect codes. In particular, there are 16 $STS(15)$s that can be well ordered and, therefore, produce 1-perfect codes: $1 - 10$ and $13 - 18$. If, for each class of nonisomorphic $STS(15)$s, we choose as representative the one

TABLE 4.1
Well-ordered STSs for $n = 15$.

STS	Ordering
1	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
2	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
3	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
4	1,2,3,4,5,6,7,8,11,9,10,12,15,13,14
5	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
6	1,2,3,4,5,6,7,8,11,9,10,12,14,13,15
7	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
8	1,2,3,4,5,6,7,8,11,13,14,9,10,12,15
9	1,2,3,4,5,6,7,8,14,9,15,10,12,11,13
10	1,2,3,4,5,6,7,8,14,10,12,9,15,11,13
13	1,2,3,4,5,6,7,8,11,13,14,9,10,12,15
14	1,2,3,4,5,6,7,8,11,13,14,9,10,12,15
15	1,2,3,4,5,6,7,8,11,12,15,9,10,13,14
16	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
17	1,2,3,4,5,6,7,8,11,13,14,9,10,12,15
18	1,2,3,4,5,6,7,8,12,9,13,10,14,11,15

given in [1], an example of well-ordering (although not the only one), calculated computationally, associated with each of these $STS(15)$s, is the one listed in Table 4.1.

In the specific case $n = 15$, we have studied the codes constructed using the well-ordering given in Table 4.1 and calculated r_e and r_n, respectively the outer rank and dimension of the kernel:

$$r_e = \min\{k | k = \dim(E),\ C \subset E,\ E \text{ is a vector space}\},$$

$$r_n = \dim(E), \text{ where } E = \{x \in C | x + C \subset C\}.$$

The results obtained, for the representatives we have chosen from each family, are as follows:

STS	1	2	3	4	5	6	7	8	9	10	13	14	15	16	17	18
r_n	11	9	8	7	8	6	8	7	6	6	6	6	6	8	6	6
r_e	11	12	13	13	13	13	13	14	14	14	14	14	14	14	14	14

For a given STS, by considering other well-orderings, we can obtain 1-perfect codes that are neither isomorphic nor equivalent amongst themselves. Thus, for each STS we obtain a family of 1-perfect codes.

For a given code C, if we consider $C + v$, where $v \in C$, we obtain another code equivalent to the first one that does not have to have the same $STS(n)$ associated with it, or, in other words, the STS_vs associated with each of the codewords $v \in C$ do not necessarily have to match (if they match, the code is known as *homogeneous*).

In general, each of these $STS(n)$s, together with a well-ordering, will result in a partition, taking as classes $C_i = \{x | x \in \boldsymbol{F}^n,$ where $\phi(x) = e_i\}$, where all the classes are 1-perfect codes (what we have called a 1-perfect partition).

5. Error-correcting, coding, and decoding. With the codes obtained, error-correcting is very easy. In essence, the codewords are characterized by having a constant value (the value map is defined in (3.3)). Therefore, when we receive a word, we can calculate its value and use it as a syndrome to correct errors.

Let us assume a code C defined using a well-ordered STS, which consists of all the vectors with value e_i, $C = \{v \in \boldsymbol{F}^n \mid \phi(v) = e_i\}$.

TABLE 5.1
Redundant bits for code 17.

ϕ	$x_{10} = 0$	$x_{10} = 1$
0	0000	0111
1	1001	1110
2	1010	1101
3	1111	1000
4	1100	1011
5	0110	0001
6	0011	0100
7	0101	0010
8	1011	1100
9	0100	0011
10	0111	0000
11	1101	1010
12	0010	0101
13	1110	1001
14	1000	1111
15	0001	0110

Given any vector $v \in \boldsymbol{F}^n$, we can compute its syndrome $\phi(v)$ and we will have $\phi(v) = e_i$ if and only if $v \in C$.

If $v \notin C$, we have $\phi(v) = e_k$, where $e_k \neq e_i$. Let e_j be such that $e_i = e_k * e_j$. Now we will calculate the only vector $w \in C$ at distance one from v as $w = v \cdot e_j$, since $d(v, w) = 1$ and $\phi(w) = \phi(v \cdot e_j) = \phi(v) * e_j = e_k * e_j = e_i$ (see Theorem 3.5).

Concerning coding-decoding, we were unable to show that, for any value of n, the codes obtained are systematic, although in the specific case $n = 15$, Table 4.1 gives systematic codes where the 11 information coordinates are $1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13$ and the 4 redundant coordinates are $11, 12, 14, 15$ (after the well-ordering).

We have not included the proof that the codes in Table 4.1 are systematic since it is only of interest in the particular case $n = 15$.

Example. We will provide an example of the above, using the code 17 defined with the order given in Table 4.1.

The STS which results in this code is formed by the following triples (see [1]):
(1, 2, 3), (1, 4, 5), (1, 6, 7), (1, 8, 9), (1, 10, 11), (1, 12, 13),
(1, 14, 15), (2, 4, 6), (2, 5, 7), (2, 8, 10), (2, 9, 11), (2, 12, 14),
(2, 13, 15), (3, 4, 7), (3, 5, 6), (3, 8, 12), (3, 9, 13), (3, 10, 14),
(3, 11, 15), (4, 8, 15), (4, 9, 14), (4, 10, 13), (4, 11, 12), (5, 8, 11),
(5, 9, 12), (5, 10, 15), (5, 13, 14), (6, 8, 14), (6, 9, 10), (6, 11, 13),
(6, 12, 15), (7, 8, 13), (7, 9, 15), (7, 10, 12), (7, 11, 14).

The codeword that we wish to construct will be v, of which we know the 11 coordinates $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_{11}, x_{13}, x_{10}$. Starting with these coordinates and using the value $\phi(v)$ we can calculate the 4 redundant symbols $x_{14}, x_9, x_{12}, x_{15}$, according to the coordinate 10 in the way described in Table 5.1.

- Let us suppose the information is given by the 11 bits 010 111 001 10, which we assume are the coordinates $1, 2, 3, 4, 5, 6, 7, 8, 11, 13, 10$ of the codeword we wish to construct (we have used the order given in Table 4.1 for code 17).
- Using the operation defined in $\boldsymbol{A}_n$ (according to the Steiner triples), we calculate $\phi = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_{11}, x_{13}] = [e_2, e_4, e_5, e_6, e_{11}, e_{13}] = e_7$.
- According to Table 5.1, for this value of $\phi(v) = e_7$ and knowing $x_{10} = 0$,

there is a redundancy 0101 for which the codeword will be

$$v = (010\,111\,001\,101\,001)$$

(the order of the coordinates is $1, 2, 3, 4, 5, 6, 7, 8, 11, 13, 14, 9, 10, 12, 15$).

- Let us assume that a transmission error has occurred and that the vector received is $w = (010\,101\,001\,101\,001)$.
- Let us calculate the syndrome for the vector received as $\phi(w) = [e_2, e_4, e_6, e_{11}, e_{13}, e_9, e_{15}] = e_5$.
- We will correct the error made by calculating the vector $v = w \cdot e_5$ since $\phi(v) = \phi(w \cdot e_5) = e_5 * e_5 = 0$;
$\phi(v) = \phi(w \cdot e_5) = \phi(w) * e_5 = [e_2, e_4, e_6, e_{11}, e_{13}, e_9, e_{15}] * e_5 =$
$[e_2, e_4, e_6, e_{11}, e_{13}, e_9, e_{15}, e_5] = [e_2, e_4, e_6, e_{11}, e_{13}, e_9, e_5, e_{15}] =$
$[e_2, e_4, e_6, e_{11}, e_{13}, e_1, e_9, e_{15}] = [e_2, e_4, e_6, e_{11}, e_5, e_{13}, e_9, e_{15}] =$
$[e_2, e_4, e_6, e_5, e_{11}, e_{13}, e_9, e_{15}] = [e_2, e_4, e_5, e_6, e_{11}, e_{13}, e_9, e_{15}]$,
so $v = (010111001101001)$.

Remark. The calculation made, $[e_{a_1} e_{a_2} \cdots e_{a_r} e_i] = [e_{a_1} e_{a_2} \cdots e_{a_{r-1}} e_{i'} e_{a_r}] =$ $[e_{a_1} e_{a_2} \ldots e_{a_{r-2}} e_{i''} e_{a_{r-1}} e_{a_r}] = \ldots$ is as described in (3.4).

6. Conclusions and further research. In this paper, we have seen that a partition of the n-cube on 1-perfect codes is equivalent to having a quasi-group structure $\boldsymbol{A}_n = \{e_0, e_1, e_2, \ldots, e_n\}$, with zero element $e_0 = 0$, which acts in a distance-compatible way on $\boldsymbol{F}^n$ and is an $\boldsymbol{A}_n$-homomorphic image of $\boldsymbol{F}^n$.

In the specific case that $\boldsymbol{A}_n$ is considered to be the structure derived from a well-ordered STS, we have seen an effective way to construct 1-perfect partitions and, therefore, 1-perfect codes, that in the case $n = 15$ are systematic. Moreover, it is not difficult to see that according to the nomenclature of Etzion and Vardy (see [2]), these 1-perfect codes are of the noninterlaced type.

Further research in this topic should include the following:

- A consideration of quasi-group structures on $\boldsymbol{A}_n$ with more characteristics, for example, commutativity or associativity. In the extreme case, analysis should also consider the case when $\boldsymbol{A}_n$ has the commutative group structure. In this situation, the factorization theorem of commutative groups indicates what the $\boldsymbol{A}_n$ algebraic structure should be like.
- A consideration of distance-compatible actions of $\boldsymbol{A}_n$ on $\boldsymbol{F}^n$, which vary from the one given by the construction included in this paper. For instance it could be interesting to construct distance-compatible actions of $\boldsymbol{A}_n$ on $\boldsymbol{F}^n$ that could not be extended the whole n-cube.
- The existence of well-ordered STSs for all n as well as is proved in the specific case $n = 15$.
- The codes obtained in this paper are systematic for any value of n as well as in the case $n = 15$.
- Characterization of the 1-perfect partitions such that we use the partition to determine the algebraic properties of the perfect codes which make it up. For example, using uniform 1-perfect partitions, we can give the propelinear structure to all the classes of the partition (see [8]).

Appendix.

STS	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_{16}
1	225	0	0	0	0	0	0	0
2	129	96	0	0	0	0	0	0
3	113	24	24	64	0	0	0	0
4	73	60	60	32	0	0	0	0
5	73	108	12	32	0	0	0	0
6	45	42	90	48	0	0	0	0
7	57	36	36	96	0	0	0	0
8	65	52	12	32	28	28	8	0
9	41	32	48	52	34	18	0	0
10	41	36	62	46	26	12	2	0
11	25	6	50	62	62	12	8	64
12	41	12	69	48	45	0	10	64
13	57	20	28	72	48	0	0	0
14	65	12	24	72	36	12	4	0
15	37	30	50	40	32	36	0	0
16	113	0	0	56	0	0	56	0
17	57	12	12	64	24	48	8	0
18	37	30	38	48	24	44	4	0
19	21	14	42	64	36	36	12	64
20	23	6	36	49	63	39	9	64
21	25	0	15	81	69	21	14	91
22	21	0	12	86	77	12	17	91
23	23	6	47	42	57	34	16	122
24	23	4	38	44	57	34	25	173
25	33	4	21	45	58	51	13	167
26	37	6	32	36	51	42	21	165
27	21	2	29	37	64	57	15	183
28	21	2	25	41	53	61	22	224
29	29	6	18	54	42	54	22	178
30	21	0	10	41	76	51	26	252
31	23	8	56	38	50	34	16	96
32	17	0	15	52	49	60	32	252
33	17	0	11	33	56	61	47	297
34	19	0	10	33	51	75	37	282
35	25	0	0	30	48	72	50	313
36	19	0	8	32	96	56	14	268
37	19	0	0	24	72	36	74	322
38	19	0	4	23	44	83	52	337
39	19	0	6	39	64	73	24	290
40	23	0	6	29	73	61	33	298

STS	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_{16}
41	23	0	3	26	62	76	35	315
42	17	0	0	17	71	79	41	368
43	21	0	3	9	105	69	18	296
44	17	0	4	19	65	77	43	354
45	17	0	7	20	61	76	44	334
46	17	0	0	14	53	88	53	373
47	17	0	9	32	56	70	41	299
48	17	0	5	23	69	73	38	317
49	17	0	2	16	58	84	48	361
50	17	0	2	26	62	88	30	300
51	19	0	1	21	58	91	35	354
52	17	0	2	25	59	83	39	344
53	19	0	3	33	63	69	38	307
54	19	0	6	34	63	68	35	311
55	17	0	6	25	57	91	29	344
56	17	0	4	23	64	79	38	333
57	17	0	3	12	70	82	41	333
58	17	0	5	40	75	56	32	259
59	17	0	6	43	66	69	24	295
60	17	0	0	21	57	81	49	364
61	15	0	0	63	77	21	49	91
62	15	0	9	26	44	90	41	328
63	15	2	12	37	48	75	36	271
64	15	0	3	24	53	66	64	337
65	15	0	3	18	57	88	44	350
66	15	0	0	15	52	93	50	374
67	15	0	0	13	60	99	38	377
68	15	0	2	18	68	80	42	359
69	15	0	2	14	47	86	61	379
70	15	0	7	28	53	88	34	338
71	15	0	4	11	59	85	51	355
72	15	0	1	17	60	87	45	360
73	15	0	0	14	50	98	48	380
74	15	0	16	32	40	98	24	289
75	15	0	3	36	63	90	18	337
76	15	0	15	25	85	45	40	330
77	15	0	0	3	33	111	63	412
78	15	0	4	26	62	98	20	340
79	15	0	18	18	72	90	12	212
80	15	0	0	0	0	90	120	455

Acknowledgments. The author wishes to thank J. Borges, K. Phelps, and J. Pujol for useful discussions and valuable comments during the preparation of this paper.

REFERENCES

[1] C.J. Colbourn and J.H. Dinitz, *The CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton, FL, 1996.
[2] T. Etzion and A. Vardy, *Perfect binary codes: Constructions, properties and enumeration*, IEEE Trans. Inform. Theory, 40 (1994), pp. 754–763.
[3] J.M. LeVan, *Designs and Codes*, Ph.D. thesis, Auburn University, Auburn, AL, 1995.
[4] F.J. MacWilliams and N.J.A. Sloane, *Error Correcting Codes*, North–Holland, New York, 1977.
[5] R.A. Mathon, K.T. Phelps, and A. Rosa, *Small triple systems and their properties*, Ars Combin., 15 (1983), pp. 3–110.
[6] H.O. Pflugfelder, *Quasigroups and Loops. Introduction*, Helderman-Verlag, Berlin, 1990.
[7] K.T. Phelps, *A combinatorial construction of perfect codes*, SIAM J. Algebraic Discrete Meth., 4 (1983), pp. 398–403.
[8] J. Rifà and J. Pujol, *Distance invariant propelinear codes*, IEEE Trans. Inform. Theory, 43 (1997), pp. 590–598.
[9] H.S. White, F.N. Cole, and L.D. Cummings, *Complete classification of the triad systems of fifteen elements*, Mem. Mat. Acad. Sci. USA, 2nd memoir, 14 (1919), pp. 1–89.

SIAM J. Discrete Math.
Vol. 12, No. 1, pp. 48–63

CONSTRUCTIVE QUASI-RAMSEY NUMBERS AND TOURNAMENT RANKING*

A. CZYGRINOW†, S. POLJAK†, and V. RÖDL†

Abstract. A constructive lower bound on the quasi-Ramsey numbers and the tournament ranking function was obtained in [S. Poljak, V. Rödl, and J. Spencer, *SIAM J. Discrete Math.*, (1) 1988, pp. 372–376]. We consider the weighted versions of both problems. Our method yields a polynomial time heuristic with guaranteed lower bound for the linear ordering problem.

Key words. discrepancy, linear ordering problem, derandomization, regularity lemma

AMS subject classifications. 68R05, 68R10, 05D99

PII. S0895480197318301

1. Introduction.

The *quasi-Ramsey number* $g(n)$ is defined as the maximum discrepancy between the number of edges and nonedges that appears on some induced subgraph of any graph of order n, i.e.,

$$g(n) = \min_{f} \max_{S \subseteq [n]} |f(S)|,$$

where $[n] = 1, \ldots, n$, f is a function from $[n]^2$ into $\{-1, 1\}$ and $f(S) = \sum_{e \in S^2} f(e)$. It is well known (Erdös and Spencer [4]) that for some positive, absolute constants c_1, c_2

$$c_1 n^{3/2} \leq g(n) \leq c_2 n^{3/2}.$$

The *tournament ranking function* $h(n)$ is defined as the maximum size of an acyclic (undirected) subgraph that appears in any tournament of order n. More precisely, let T_n be a tournament on n vertices, P_n a transitive tournament on n vertices, and let $|T_n \cap P_n|$ denote the number of common oriented arcs of T_n and P_n; then

$$h(n) = \min_{T_n} \max_{P_n} |T_n \cap P_n|.$$

It was shown by Spencer ([14], [15]) that

$$\frac{1}{2}\binom{n}{2} + c_1 n^{3/2} \leq h(n) \leq \frac{1}{2}\binom{n}{2} + c_2 n^{3/2}$$

where c_1 and c_2 are positive absolute constants. The proof of the upper bound has been simplified by Fernandez de La Vega [5]. Using the method of Spencer, the lower bound on $h(n)$ can be obtained by an algorithmic argument from the lower bound on $g(n)$.

Poljak, Rödl, and Spencer [12] proposed a fast $O(n^3 \log n)$ time algorithm that finds a set S with discrepancy at least $\frac{\pi^{-1/2}}{24} n^{3/2}$, the corresponding result for the tournament ranking function $h(n)$ is also presented in [12]. We will consider the

*Received by the editors March 12, 1997; accepted for publication (in revised form) March 5, 1998; published electronically January 29, 1999.
http://www.siam.org/journals/sidma/12-1/31830.html

†Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322 (aczygri@mathcs.emory.edu, rodl@mathcs.emory.edu).

weighted version of both problems. Our algorithm uses the Erdös–Selfridge method of conditional expectations that was also applied in [12]. For the lower bound on the quasi-Ramsey number $g(n)$ we prove the following result.

THEOREM 1. *Let $f : [n]^2 \to \mathbb{R}$ be a weight function on the edges of a complete graph K_n. Then there is a subset $S \subset [n]$ such that*

$$|f(S)| \geq \frac{1}{12\sqrt{\pi}} n^{-1/2} \sum_{e \in [n]^2} |f(e)|.$$

Moreover, S can be found in $O(n^3 \lg(nd) \lg n)$ time, provided the weights are integers from $\{-d, \ldots, d\}$.

The weighted version of the tournament ranking problem is also known as the *linear ordering problem* (see Grötschel, Jünger, and Reinelt [10]). The problem can be formulated in the following way: For a given tournament T with weight $c(i, j)$ on the arc $(i, j) \in T$, find the ordering σ of vertices for which the sum

$$\sum_{(i,j) \in T, \sigma(i) < \sigma(j)} c(i, j)$$

is a maximum. The list of applications of the linear ordering problem can be found in Lenstra [11]. It includes applications from different areas of econometrics (input-output matrix analysis), sociology (social choice), psychology, machine scheduling, and even archaeology. The problem is NP-complete (see Garey and Johnson [8]), but there were several methods developed for solving small instances, e.g., up to order of 50 by Grötschel, Jünger, and Reinelt [10]. Using the algorithm from Theorem 1, we will get a polynomial time heuristic with a guaranteed lower bound.

THEOREM 2. *Let T be a tournament on n vertices with nonnegative weights $w(e)$ on edges. Then there is an ordering σ such that the sum of weights on edges that agree with the ordering is at least*

$$\left(\frac{1}{2} + \frac{1}{4\sqrt{\pi}} n^{-1/2} \right) K,$$

where K is the total sum of weights. The ordering σ can be constructed by a $O(n^3 \lg (nd) \lg n)$ time algorithm, provided weights are integers from $\{0, \ldots, d\}$.

From the upper bound on $h(n)$, we conclude that there exists weight function for which the heuristic is best possible (up to a constant factor).

Given a real number ρ, $0 < \rho < 1$ a polynomial time approximation scheme (PTAS) for an optimization problem is an algorithm which when given an instance of size n, finds in polynomial time (in n) a solution of value at least $(1 - \rho)OPT$, where OPT is the optimal value. Using the regularity lemma and its constructive version of Alon et al. [1], we design a PTAS for the "dense" quasi-Ramsey problem and for tournament ranking. For the quasi-Ramsey number we have the following theorem. Let $f : E(K_n) \to \{-1, 1\}$ and $OPT(f) = \max_{S \subseteq [n]} |f(S)|$.

THEOREM 3. *Let $c > 0$ be fixed. If $OPT(f) \geq cn^2$, then for every ρ, $0 < \rho < 1$, there is a $O(n^{2.4})$ time algorithm that constructs set S such that*

$$|f(S)| \geq (1 - \rho)OPT(f).$$

For the tournament ranking we prove the following theorem for the case when $OPT(T_n) = \max_{P_n} |T_n \cap P_n|$ for a tournament T_n.

THEOREM 4. *For $0 < \rho < 1$ there is a polynomial time algorithm that constructs an ordering σ of vertices of T_n so that at least $(1-\rho)OPT(T_n)$ of arcs agree with σ.*

Note that Theorem 3 and Theorem 4 are in some sense counterparts to Theorem 1 and Theorem 2. For example, Theorem 1 provides the existence of a polynomial time algorithm to find the set S with $|f(S)|$ being the guaranteed minimum; Theorem 3 gives for every ρ the const$(\rho)n^{2.4}$ algorithm that finds a set S with $f(S)$ being a $(1-\rho)$ multiple of the optimal. Theorem 3 is based on the algorithmic version of the regularity lemma which "approximates the graph with error of ϵn^2". Therefore, it can be applied only to instances with $OPT(f) \geq cn^2$. On the other hand, in case of Theorem 4, clearly $OPT(T_n) \geq \frac{1}{2}\binom{n}{2}$ and, therefore, a PTAS for the linear ordering problem can be obtained with no additional assumptions. Independently, very recently Frieze and Kannan [6] and [7] applied a version of the regularity lemma to the maximum subgraph problem, an equivalent to tournament ranking. Our arguments differ from those in [7]. The rest of the paper is organized as follows: In section 2, for a given $\overrightarrow{v_1}, \ldots, \overrightarrow{v_n} \in \mathbb{R}^k$, we will show how to construct sign vector $\overrightarrow{X} = (X_1, \ldots, X_n)$ such that

$$\| X_1\overrightarrow{v_1} + \cdots + X_n\overrightarrow{v_n} \| \geq cn^{-1/2} \sum_{i=1}^{n} \| \overrightarrow{v_i} \|,$$

where $\|\overrightarrow{u}\| = \sum_{j=1}^{k} |u_j|$. The algorithm is later applied to quasi-Ramsey numbers and to the linear ordering problem. Section 3 includes the applications of the regularity lemma. We conclude with an open problem in section 4.

2. Constructing sign vectors. Set $\overrightarrow{1} = (1, \ldots, 1)$ and $\overrightarrow{0} = (0, \ldots, 0)$, and for $\overrightarrow{u}$ and $\overrightarrow{v}$ from $\mathbb{R}^k$, let $\langle \overrightarrow{v}, \overrightarrow{u} \rangle$ denote the dot product of $\overrightarrow{v}$ and $\overrightarrow{u}$, and $\|\overrightarrow{u}\|$ its l_1- norm, i.e., $\|\overrightarrow{u}\| = \sum_{j=1}^{k} |u_j|$. We first establish two auxiliary facts.

LEMMA 5.

$$\sum_{\overrightarrow{X} \in \{-1,1\}^n} |\langle \overrightarrow{1}, \overrightarrow{X} \rangle| = 2n \binom{n-1}{\lfloor \frac{n}{2} \rfloor}.$$

The proof can be found in [12]. For $1 \leq i \leq n$, let $\mathbf{X}_i$ be independent random variables with distribution $Pr(\mathbf{X}_i = 1) = Pr(\mathbf{X}_i = -1) = \frac{1}{2}$.

LEMMA 6. *Let $b_1, \ldots, b_n$ and a be real numbers and let u be the arithmetic mean of $|b_1|, \ldots, |b_n|$. Then we have the following inequality:*

$$E(|a + \mathbf{X}_1 b_1 + \cdots + \mathbf{X}_n b_n|) \geq E(|a + \mathbf{X}_1 u + \cdots + \mathbf{X}_n u|).$$

Proof. We may assume that all b_i's are nonnegative since the random variables $Z_i = \mathrm{sgn(b_i)X_i}$ have the same distribution as X_i, i.e., $E(|a + X_1 b_1 + \cdots + X_n b_n|) = E(|a + Z_1 b_1 + \cdots + Z_n b_n|) = E(|a + X_1 |b_1| + \cdots + X_n |b_n||)$. Given a vector $\overrightarrow{w} = (w_1, \ldots, w_n) \in \mathbb{R}^n$, let $\overrightarrow{w^{(l)}}$ be the vector obtained from $\overrightarrow{w}$ by cyclic shifting, with ith coordinate $w_i^{(l)} = w_{i+l \mod n}$ for $i = 1, 2, \ldots, n$. We have

$$E(|a+\mathbf{X}_1 b_1+\cdots+\mathbf{X}_n b_n|) = \frac{1}{2^n} \sum_{\overrightarrow{X} \in \{-1,1\}^n} \left| a + \sum_{i=1}^{n} X_i b_i \right| = \frac{1}{2^n} \sum_{\overrightarrow{X}} \frac{1}{n} \sum_{l=1}^{n} \left| a + \sum_{i=1}^{n} X_i^{(l)} b_i \right|$$

$$\geq \frac{1}{2^n}\sum_{\overrightarrow{X}}\frac{1}{n}\left|na+\sum_{i=1}^{n}\sum_{l=1}^{n}X_i^{(l)}b_i\right| = \frac{1}{2^n}\sum_{\overrightarrow{X}}\left|a+\sum_{i=1}^{n}X_iu\right| = E(|a+\mathbf{X}_1u+\cdots+\mathbf{X}_nu|). \qquad \square$$

Lemma 7. *Let* $\overrightarrow{v_1},\ldots,\overrightarrow{v_n}\in\mathbb{R}^k$. *Then*

$$E(\|\mathbf{X}_1\overrightarrow{v_1}+\cdots+\mathbf{X}_n\overrightarrow{v_n}\|)\geq\sqrt{\frac{2}{\pi}}n^{-1/2}\sum_{i=1}^{n}\|\overrightarrow{v_i}\|.$$

Proof. From Lemma 5 and Stirling's formula, we obtain

$$E(|\mathbf{X}_1+\cdots+\mathbf{X}_n|)=\frac{1}{2^n}\sum_{\overrightarrow{X}\in\{-1,1\}^n}|\langle\overrightarrow{1},\overrightarrow{X}\rangle| = 2n2^{-n}\binom{n-1}{\lfloor\frac{n}{2}\rfloor}\geq\sqrt{\frac{2n}{\pi}}.$$

Let u_j be the arithmetic mean of absolute values of the jth components of $\overrightarrow{v_1},\ldots,\overrightarrow{v_n}$, where $j=1,\ldots,k$ and let $\overrightarrow{u}=(u_1,\ldots,u_k)$. Using Lemma 6 with $a=0$ we have

$$E(\|\mathbf{X}_1\overrightarrow{v_1}+\cdots+\mathbf{X}_n\overrightarrow{v_n}\|)\geq E(\|\mathbf{X}_1\overrightarrow{u}+\cdots+\mathbf{X}_n\overrightarrow{u}\|)=\sum_{j=1}^{k}E(|\mathbf{X}_1u_j+\cdots+\mathbf{X}_nu_j|)$$

$$=\sum_{j=1}^{k}u_jE(|\mathbf{X}_1+\cdots+\mathbf{X}_n|)\geq\sqrt{\frac{2}{\pi}}n^{-1/2}\sum_{i=1}^{n}\|\overrightarrow{v_i}\|. \qquad \square$$

Corollary 8. *For given* $\overrightarrow{v_1},\ldots,\overrightarrow{v_n}\in\mathbb{R}^k$, *there is a choice of signs* $(X_1,\ldots,X_n)\in\{-1,1\}^n$ *such that*

$$\|X_1\overrightarrow{v_1}+\cdots+X_n\overrightarrow{v_n}\|\geq\sqrt{\frac{2}{\pi}}n^{-1/2}\sum_{i=1}^{n}\|\overrightarrow{v_i}\|.$$

Next we will show that a vector $\overrightarrow{X}=(X_1,\ldots,X_n)$ from Corollary 8 can be constructed by a polynomial time algorithm. The idea is as follows. We have $E(\|\mathbf{X}_1\overrightarrow{v_1}+\cdots+\mathbf{X}_n\overrightarrow{v_n}\|)\geq T$, where $T=cn^{-1/2}\sum\|\overrightarrow{v_i}\|$ in the beginning. (For later convenience, we write the vectors in the reverse order.) Let us assume that signs $X_n,X_{n-1},\ldots,X_{i+1}$ are chosen, one in each step, such that

$$E(\|X_n\overrightarrow{v_n}+\cdots+X_{i+1}\overrightarrow{v_{i+1}}+\mathbf{X}_i\overrightarrow{v_i}+\cdots\mathbf{X}_1\overrightarrow{v_1}\|)\geq T.$$

At this moment there are two possible choices of X_i, and we take the better one (the one that maximizes the value of the expectation). As we cannot compute quickly the expected value $E(\|X_n\overrightarrow{v_n}+\cdots+X_{i+1}\overrightarrow{v_{i+1}}+\mathbf{X}_i\overrightarrow{v_i}+\cdots+\mathbf{X}_1\overrightarrow{v_1}\|)$ for general $\overrightarrow{v_i},\ldots,\overrightarrow{v_1}$, we compute $E(\|X_n\overrightarrow{v_n}+\cdots+X_{i+1}\overrightarrow{v_{i+1}}+\mathbf{X}_i\overrightarrow{u}+\cdots+\mathbf{X}_1\overrightarrow{u}\|)$ instead, where $\overrightarrow{u}$ is the component-wise "average" of $\overrightarrow{v_1},\ldots,\overrightarrow{v_n}$.

To describe the algorithm more precisely, we need to introduce some notation. For vectors $\overrightarrow{a}=(a_1,\ldots,a_k)$, $\overrightarrow{b}=(b_1,\ldots,b_k)\in\mathbb{R}^k$ we define the polynomials

$$W(b_j,i,a_j)=E(|b_j+\mathbf{X}_ia_j+\cdots+\mathbf{X}_1a_j|)=\sum_{l=0}^{i}\binom{i}{l}2^{-i}|b_j+a_j(i-2l)|,$$

$$W(\overrightarrow{b}, i, \overrightarrow{a}) = \sum_{j=1}^{k} W(b_j, i, a_j) = \sum_{j=1}^{k} \sum_{l=0}^{i} \binom{i}{l} 2^{-i} |b_j + a_j(i-2l)|.$$

For given $\overrightarrow{v_i} = (v_{i1}, \dots, v_{ik}) \in \mathbb{R}^k$, $i = 1, \dots, n$, let u_{ij} denote the arithmetic mean of absolute values of the jth coordinates of $\overrightarrow{v_i}, \dots, \overrightarrow{v_1}$, i.e., $u_{ij} = \frac{1}{i}(|v_{ij}| + \cdots + |v_{1j}|)$, and set $\overrightarrow{u_i} = (u_{i1}, \dots, u_{ik})$. By $\overrightarrow{S_i}$ we denote the partial sums: let $\overrightarrow{S_n} = \overrightarrow{0}$ and $\overrightarrow{S_i} = X_n \overrightarrow{v_n} + \cdots + X_{i+1} \overrightarrow{v_{i+1}}$, where $X_n, \dots, X_{i+1}$ have already been defined. (Observe that $E(\|\overrightarrow{S_i} + \mathbf{X}_i \overrightarrow{u_i} + \dots \mathbf{X}_1 \overrightarrow{u_i}\|) = W(\overrightarrow{S_i}, i, \overrightarrow{u_i})$.) Now we choose

$$X_i = \begin{cases} 1 & \text{if } W(\overrightarrow{S_i} + \overrightarrow{v_i}, i-1, \overrightarrow{u_{i-1}}) \geq W(\overrightarrow{S_i} - \overrightarrow{v_i}, i-1, \overrightarrow{u_{i-1}}), \\ -1 & \text{otherwise.} \end{cases}$$

We can formalize the algorithm in the following procedure.

ALGORITHM
input: vectors $\overrightarrow{v_1}, \dots, \overrightarrow{v_n} \in \mathbb{R}^k$
output: sign vector $(X_1, \dots, X_n)$
$\overrightarrow{S_n} = \overrightarrow{0}$
for i=n downto 1
begin
if $i < n$ then $\overrightarrow{S_i} = X_n \overrightarrow{v_n} + \cdots + X_{i+1} \overrightarrow{v_{i+1}}$
compute $W_+ = W(\overrightarrow{S_i} + \overrightarrow{v_i}, i-1, \overrightarrow{u_{i-1}})$ and $W_- = W(\overrightarrow{S_i} - \overrightarrow{v_i}, i-1, \overrightarrow{u_{i-1}})$
if $W_+ \geq W_-$ then $X_i = 1$
else $X_i = -1$
end
return $(X_1, \dots, X_n)$

PROPOSITION 9. *The above algorithm returns a vector* $(X_1, \dots, X_n)$ *such that*

$$\|X_1 \overrightarrow{v_1} + \cdots + X_n \overrightarrow{v_n}\| \geq \sqrt{\frac{2}{\pi}} n^{-1/2} \sum_{i=1}^{n} \|\overrightarrow{v_i}\|.$$

Proof. Since $E(\|\overrightarrow{S_i} + \mathbf{X}_i \overrightarrow{v_i} + \mathbf{X}_{i-1} \overrightarrow{u_{i-1}} + \cdots \mathbf{X}_1 \overrightarrow{u_{i-1}}\|) = \frac{1}{2} W(\overrightarrow{S_i} + \overrightarrow{v_i}, i-1, \overrightarrow{u_{i-1}}) + \frac{1}{2} W(\overrightarrow{S_i} - \overrightarrow{v_i}, i-1, \overrightarrow{u_{i-1}})$, we have

$$\begin{aligned} W(\overrightarrow{S_{i-1}}, i-1, \overrightarrow{u_{i-1}}) &= W(\overrightarrow{S_i} + X_i \overrightarrow{v_i}, i-1, \overrightarrow{u_{i-1}}) \geq E(\|\overrightarrow{S_i} + \mathbf{X}_i \overrightarrow{v_i} + \mathbf{X}_{i-1} \overrightarrow{u_{i-1}} \\ &\quad + \cdots + \mathbf{X}_1 \overrightarrow{u_{i-1}}\|) \\ &\geq E(\|\overrightarrow{S_i} + \mathbf{X}_i \overrightarrow{u_i} + \cdots \mathbf{X}_1 \overrightarrow{u_i}\|) = W(\overrightarrow{S_i}, i, \overrightarrow{u_i}). \end{aligned}$$

The first inequality holds by the choice of X_i, the second one by Lemma 6, and the (obvious) fact that u_{ij} is an arithmetic mean of v_{ij} and $i-1$ copies of u_{i-1j}. Hence

$$\|X_n \overrightarrow{v_n} + \cdots + X_1 \overrightarrow{v_1}\| \geq W(\overrightarrow{S_1}, 1, \overrightarrow{u_1}) \geq \cdots \geq W(\overrightarrow{S_n}, n, \overrightarrow{u_n})$$

and

$$\begin{aligned} W(\overrightarrow{S_n}, n, \overrightarrow{u_n}) &= \sum_{j=0}^{k} \sum_{l=0}^{n} \binom{n}{l} 2^{-n} |u_{nj}(n-2l)| = \sum_{j=0}^{k} u_{nj} 2^{-n} \sum_{l=0}^{n} \binom{n}{l} |n-2l| \\ &\geq \sqrt{\frac{2}{\pi}} n^{1/2} \sum_{j=0}^{k} u_{nj} = \sqrt{\frac{2}{\pi}} n^{-1/2} \sum_{i=1}^{n} \|\overrightarrow{v_i}\|. \qquad \square \end{aligned}$$

PROPOSITION 10. *For $k = O(n)$, the time complexity of the above algorithm is $O(n^3 \lg(nd) \lg n)$ provided the vectors $\overrightarrow{v_1}, \ldots, \overrightarrow{v_n} \in \mathbb{R}^k$ are integral and $|v_{ij}| \leq d$.*

Proof. The procedure consists of n iterations for computing $X_n, \ldots, X_1$. At each step we evaluate the expression $W(\overrightarrow{S_i}, i, \overrightarrow{u_i})$. To keep the computation in integers we replace it by

$$i2^i W(\overrightarrow{S_i}, i, \overrightarrow{u_i}) = \sum_{l=0}^{i} \binom{i}{l} \bigg(\sum_{j=1}^{k} |iS_{ij} + (i-2l)iu_{ij}| \bigg),$$

where $\overrightarrow{S_i} = (S_{i1}, \ldots, S_{ik})$. The $O(n^2)$ combinatorial coefficients $\binom{i}{l}$ can be evaluated in advance using the identity $\binom{i}{l} = \binom{i-1}{l} + \binom{i-1}{l-1}$. Since i is of size at most n and the terms S_{ij}, iu_{ij} are of size nd, we can compute $|iS_{ij} + (i-2l)iu_{ij}|$ in $O(\lg n \lg(nd))$ steps. The sum $\sum_{j=1}^{k} |iS_{ij} + (i-2l)iu_{ij}|$ can be evaluated in $O(k \lg n \lg(nd))$ steps. The number $\binom{i}{l}$ is less than 2^n and so the multiplication $\binom{i}{l} \cdot (\sum_{j=1}^{k} |iS_{ij} + (i-2l)iu_{ij}|)$ can be computed in $O(\lg(2^n) \lg(ndk))$ steps. The total complexity of the procedure is $O(n^2(k \lg n \lg(nd) + n \lg(ndk)))$, which when $k = O(n)$ becomes $O(n^3 \lg (nd) \lg n)$. □

Using the divide and conquer technique, one can design a slightly faster algorithm that gives a little worse results (for details consult [2]).

We will now apply the algorithm to quasi-Ramsey numbers and to the linear ordering problem. Let us start with the proof of Theorem 1.

Proof of the Theorem 1. We use the same technique that was applied in [12]. Let $K = \sum_{e \in [n]^2} |f(e)|$. First we need to find a large cut of K_n with edge weights $|f(e)|$. Obviously, by a greedy procedure we can construct disjoint sets X and Y such that $X \cup Y = [n]$ and

$$\sum_{x \in X, y \in Y} |f(x,y)| \geq \frac{K}{2}.$$

Indeed, assume that sets $X^i \cup Y^i = [i]$ have been constructed. Let $W_X^i = \sum_{j \in X^i} f(j, i+1)$ and $W_Y^i = \sum_{j \in Y^i} f(j, i+1)$. If $W_X^i \leq W_Y^i$ then set $X^{i+1} = X^i \cup \{i+1\}$ and $Y^{i+1} = Y^i$; otherwise, set $X^{i+1} = X^i$ and $Y^{i+1} = Y^i \cup \{i+1\}$. (Using the Goemans–Williamson algorithm from [9], one can possibly improve a constant in our theorem. However, since the result in [9] provides .878 approximation of the optimal cut, it does not guarantee that the produced cut is bigger than $\frac{K}{2}$. For slightly better cut algorithms consult [13].)

Let $X = \{x_1, \ldots, x_{n_1}\}$, $Y = \{y_1, \ldots, y_{n_2}\}$. We assume $n_1 \leq n/2$. Assign a vector $\overrightarrow{v_i} = (v_{i1}, \ldots, v_{in_2})$ to each vertex x_i, where $v_{ij} = f(x_i, y_j)$, $i = 1, \ldots, n_1$, $j = 1, \ldots, n_2$. Using the algorithm from section 2, we construct a sign vector $(X_1, \ldots, X_{n_1})$ such that

$$\|X_1 \overrightarrow{v_1} + \cdots + X_{n_1} \overrightarrow{v_{n_1}}\| \geq \sqrt{\frac{2}{\pi}} {n_1}^{-1/2} \frac{K}{2} \geq \sqrt{\frac{2}{\pi}} (\frac{n}{2})^{-1/2} \frac{K}{2} \geq \frac{1}{\sqrt{\pi}} n^{-1/2} K.$$

We partition sets $X = X^+ \cup X^-$ and $Y = Y^+ \cup Y^-$ by $X^+ = \{x_i, X_i = 1\}$,

$Y^+ = \{y_j, \sum_{i=1}^{n_1} X_i f(x_i, y_j) \geq 0\}$ and $X^- = X - X^+$, $Y^- = Y - Y^+$. Then

$$\|X_1 \overrightarrow{v_1} + \cdots + X_{n_1} \overrightarrow{v_{n_1}}\| = \sum_{j=1}^{n_2} |\sum_{i=1}^{n_1} X_i f(x_i, y_j)| = \sum_{y \in Y^+, x \in X^+} f(x, y)$$
$$+ \sum_{y \in Y^+, x \in X^-} -f(x, y) + \sum_{y \in Y^-, x \in X^+} -f(x, y) + \sum_{y \in Y^-, x \in X^-} f(x, y).$$

Hence, we can choose $X^* \in \{X^+, X^-\}$ and $Y^* \in \{Y^+, Y^-\}$ such that

$$|f(X^*, Y^*)| = \left| \sum_{y \in Y^*, x \in X^*} f(x, y) \right| \geq \frac{1}{4\sqrt{\pi}} n^{-1/2} K.$$

We also have $f(X^*, Y^*) = f(X^* \cup Y^*) - f(X^*) - f(Y^*)$. Let S be one of X^*, Y^*, $X^* \cup Y^*$ for which $|f(S)| \geq \frac{1}{3}|f(X^*, Y^*)|$. We see that S is such that $|f(S)| \geq \frac{1}{12\sqrt{\pi}} n^{-1/2} K$. ☐

Taking $K = \binom{n}{2}$ we obtain a lower bound on the quasi-Ramsey numbers.

COROLLARY 11.

$$g(n) \geq \frac{1}{24\sqrt{\pi}} n^{3/2}.$$

We can now apply the result of Theorem 1 to the linear ordering problem. Since the proof resembles the reasoning for the corresponding result in [12], we omit the details.

Proof of Theorem 2. Let w_{ij} be the weight of the pair $\{i, j\}$. Define $f : [n]^2 \to \mathbb{Z}$ as follows. For $i < j$,

$$f(i, j) = \begin{cases} w_{ij} & \text{if } (i, j) \in T, \\ -w_{ij} & \text{if } (j, i) \in T. \end{cases}$$

Let X^*, Y^* be the sets constructed in the proof of Theorem 1 and let $R = [n] - X^* - Y^*$. Construct $\prec$ in the following way. Construct ranking on X^* such that at least half of the arcs with both endpoints in X^* are consistent with the ranking. (This can be obtained by considering an arbitrary ordering and its inverse.) Similarly construct rankings of Y^* and R. Assume that $f(X^*, Y^*) \geq 0$; then for $x \in X^*$ and $y \in Y^*$ let $x \prec y$. Suppose that $f(X^* \cup Y^*, R) \geq 0$; then for $r \in R$ and $z \in X^* \cup Y^*$ let $z \prec r$. ☐

3. Applications of the regularity lemma. In this section we present the applications of the regularity lemma to both quasi-Ramsey and tournament ranking functions. A variant of the regularity lemma was applied for max-cut, graph bisection, and a quadratic assignment problem in Frieze and Kannan [6] and for computing frequencies in graphs in Duke, Lefmann, and Rödl [3]. For simplicity, we restrict our discussion to the unweighted case, but similar results can be obtained for weighted graphs and tournaments. Let (V, E) be a graph on n vertices, for $V_1, V_2 \subset V$, $V_1 \cap V_2 = \emptyset$, the density $d(V_1, V_2)$ is defined as $d(V_1, V_2) = \frac{e(V_1, V_2)}{|V_1||V_2|}$, where $e(V_1, V_2)$ denotes the number of edges between V_1 and V_2.

DEFINITION 12. *A pair of subsets* (V_1, V_2) *is called* ϵ*-regular if for every* $W_1 \subset V_1$, *with* $|W_1| \geq \epsilon|V_1|$ *and for every* $W_2 \subset V_2$, *with* $|W_2| \geq \epsilon|V_2|$

$$|d(W_1, W_2) - d(V_1, V_2)| < \epsilon.$$

DEFINITION 13. *A partition $V_1 \cup V_1 \cup \cdots \cup V_k$ of V is ϵ-regular if*

(i) $||V_i| - |V_j|| \le 1$ *for all* i, j *and*

(ii) *all except at most* $\epsilon\binom{k}{2}$ *pairs* (V_i, V_j) *are* ϵ*-regular*

Let us now state the powerful regularity lemma of Szemerédi [16].

LEMMA 14. *For every $\epsilon > 0$ and every integer l there exist N and L such that any graph with at least N vertices admits an ϵ-regular partition $V_1 \cup \cdots \cup V_k$ with $l \le k \le L$.*

The following version can be easily concluded from the original proof [16].

LEMMA 15. *For every $\epsilon > 0$ and every integer l, there exists an N such that for any graph with at least $N = N(l, \epsilon)$ vertices and any partition P of the graph into m subsets, there exists $L = L(l, \epsilon, m)$ and an ϵ-regular partition $V_1 \cup \cdots \cup V_k$ with $l \le k \le L$ which is a refinement of P.*

The partition postulated in both lemmas can be found in $O(n^{2.4})$ time using the algorithm described in Alon et al. [1].

Proof of Theorem 3. The algorithm is as follows: Let $\epsilon = \frac{\rho c}{7}$.

1. Find an ϵ-regular partition $V_1 \cup \cdots \cup V_k$ with $k \ge \frac{1}{\epsilon}$ of the graph $G_1 = (V, f^{-1}(1))$.
2. Check all 2^k subsets of V of the form $S = \bigcup_{l \in L} V_l$, where $L \subset [k]$ and choose S that maximizes $|\sum_{1 \le i < j \le k}(2d_{ij} - 1)|V_i \cap S||V_j \cap S||$.

Note that if (V_i, V_j) is ϵ-regular with density d_{ij} in $G_1 = (V, f^{-1}(1))$, then (V_i, V_j) is ϵ-regular with density $1 - d_{ij}$ in $G_{-1} = (V, f^{-1}(-1))$. Given the partition $V_1 \cup \cdots \cup V_k$, we define $f^* : 2^{[n]} \to \mathbb{R}$ in the following way. For $T \subset [n]$, $f^*(T) = \sum_{1 \le i < j \le k}(2d_{ij} - 1)|V_i \cap T||V_j \cap T|$, where $d_{ij} = d(V_i, V_j)$.

FACT 16. *Let T^* be a minimal set that maximizes f^*. Then for every l such that $V_l \cap T^* \ne \emptyset$ the sum $\sum_{j \ne l}(2d_{lj} - 1)|V_j \cap T^*| > 0$.*

Proof. We use proof by contradiction. Assume that there exists l such that $V_l \cap T^* \ne \emptyset$ and $\sum_{j \ne l}(2d_{lj} - 1)|V_j \cap T^*| \le 0$. Then

$$f^*(T^*) = \sum_{1 \le i < j \le k}(2d_{ij} - 1)|V_i \cap T^*||V_j \cap T^*| = \sum_{j \ne l}(2d_{lj} - 1)|V_l \cap T^*||V_j \cap T^*|$$

$$+ \sum_{i,j \ne l, i<j}(2d_{ij} - 1)|V_i \cap T^*||V_j \cap T^*| = |V_l \cap T^*| \sum_{j \ne l}(2d_{lj} - 1)|V_j \cap T^*|$$

$$+ \sum_{i,j \ne l, i<j}(2d_{ij} - 1)|V_i \cap T^*||V_j \cap T^*| \le \sum_{i,j \ne l, i<j}(2d_{ij} - 1)|V_i \cap T^*||V_j \cap T^*|$$

$$= \sum_{1 \le i < j \le k}(2d_{ij} - 1)|V_i \cap (T^* \setminus V_l)||V_j \cap (T^* \setminus V_l)| = f^*(T^* \setminus V_l)$$

and we get the contradiction with minimality of T^*. ∎

FACT 17. *Let T^* be a minimal set that maximizes f^*. If $T^* \cap V_l \ne \emptyset$, then $V_l \subset T^*$.*

Note that Fact 17 implies that if S is a set found by the algorithm, then $|f^*(S)| \ge f^*(T^*)$ as the algorithm checks all the possible unions of V_i's to maximize $|f^*|$. In the same way, one can show that $|f^*(S)| \ge -f^*(L^*)$ where L^* maximizes $-f^*$.

Proof.

$$f^*(T^*) = |V_l \cap T^*| \sum_{j\neq l}(2d_{lj}-1)|V_j \cap T^*| + \sum_{i,j\neq l, i<j}(2d_{ij}-1)|V_i\cap T^*||V_j \cap T^*|$$

$$\leq |V_l| \sum_{j\neq l}(2d_{lj}-1)|V_j \cap T^*| + \sum_{i,j\neq l, i<j}(2d_{ij}-1)|V_i\cap T^*||V_j \cap T^*|$$

$$= \sum_{1\leq i<j\leq k}(2d_{ij}-1)|V_i \cap (T^*\cup V_l)||V_j\cap (T^*\cup V_l)| = f^*(T^*\cup V_l).$$

Hence, $f^*(T^*) \leq f^*(T^*\cup V_l)$ and the equality holds only if $|V_l \cap T^*| = |V_l|$ as $\sum_{j\neq l}(2d_{lj}-1)|V_j\cap T^*| > 0$ by the previous fact. ∎

It will be convenient to use the following notation. For two functions $A(n)$ and $B(n)$, we write $A(n) =_\epsilon B(n)$ if $|A(n)-B(n)| \leq \epsilon n^2$ for n large enough.

Our main lemma shows that f^* is a "good" approximation for the discrepancy function f.

LEMMA 18. *For every $U \subset V$ $|f^*(U) - f(U)| < \frac{7}{2}\epsilon n^2$.*

Proof. We divide the proof into five claims.

CLAIM 19. $f(U) =_{\frac{\epsilon}{2}} \sum_{\{i,j\}\in[k]^2} f(V_i\cap U, V_j \cap U)$.

Indeed, since $|V_i| \leq \frac{n}{k}$ and also $|V_i \cap U| \leq \frac{n}{k}$, we infer that $|f(V_i\cap U)| \leq \binom{\frac{n}{k}}{2} \leq \frac{n^2}{2k^2}$. Therefore,

$$|f(U) - \sum_{\{i,j\}\in[k]^2} f(V_i\cap U, V_j\cap U)| = \left|\sum_{i=1}^k f(V_i \cap U)\right| \leq \sum_{i=1}^k |f(V_i\cap U)| \leq \frac{n^2}{2k} \leq \frac{\epsilon}{2}n^2$$

which proves Claim 19.

We partition $[k]^2 = S \cup I \cup R$ as follows: $\{i,j\} \in S$ if and only if either $|V_i \cap U| < \epsilon|V_i|$ or $|V_j \cap U| < \epsilon |V_j|$, $\{i,j\}\in I$ if and only if the pair (V_i,V_j) is not ϵ-regular, $R = [k]^2 \setminus (S\cup I)$.

CLAIM 20. $f(U) =_\epsilon \sum_{R\cup I} f(V_i \cap U, V_j\cap U)$.

$$\left|\sum_{[k]^2} f(V_i\cap U, V_j\cap U) - \sum_{R\cup I} f(V_i \cap U, V_j \cap U)\right| = \left|\sum_S f(V_i\cap U, V_j\cap U)\right|$$

$$\leq \sum_S |f(V_i \cap U, V_j\cap U)| \leq \sum_S |V_i\cap U||V_j\cap U| \leq \binom{k}{2}\epsilon\frac{n^2}{k^2} = \frac{\epsilon}{2}n^2.$$

Since $|f(U) - \sum_{[k]^2} f(V_i\cap U, V_j \cap U)| \leq \frac{\epsilon}{2}$ by Claim 19, we infer that $|f(U) - \sum_{R\cup I} f(V_i \cap U, V_j\cap U)| \leq \epsilon$.

CLAIM 21. $f(U) =_{\frac{3\epsilon}{2}} \sum_R f(V_i\cap U, V_j\cap U)$.

Indeed, there are at most $\epsilon\frac{k^2}{2}$ irregular pairs and for each of them $|f(V_i \cap U, V_j \cap U)| \leq (\frac{n}{k})^2$, which implies

$$\left|\sum_{R\cup I} f(V_i\cap U, V_j\cap U) - \sum_R f(V_i\cap U, V_j \cap U)\right| = \left|\sum_I f(V_i \cap U, V_j\cap U)\right|$$

$$\leq \sum_I |f(V_i \cap U, V_j \cap U)| \leq \epsilon \frac{k^2}{2}\left(\frac{n}{k}\right)^2 = \frac{\epsilon}{2} n^2.$$

Together with Claim 20, this shows that $f(U) =_{\frac{3\epsilon}{2}} \sum_R f(V_i \cap U, V_j \cap U)$.

CLAIM 22. $f(U) =_{\frac{5\epsilon}{2}} \sum_R (2d_{ij}-1)|U \cap V_i||U \cap V_j|$.

From Claim 21 we know that $f(U) =_{\frac{3\epsilon}{2}} \sum_R f(U \cap V_i, U \cap V_j)$. For $\{i,j\} \in R$ we can approximate $f(U \cap V_i, U \cap V_j)$ by $(2d_{ij}-1)|U \cap V_i||U \cap V_j|$ with $2\epsilon(\frac{n}{k})^2$ error, namely,

$$\begin{aligned}
&|f(U \cap V_i, U \cap V_j) - (2d_{ij}-1)|U \cap V_i||U \cap V_j||\\
&= |d(U \cap V_i, U \cap V_j)|U \cap V_i||U \cap V_j| - (1 - d(U \cap V_i, U \cap V_j))|U \cap V_i||U \cap V_j|\\
&\quad -(2d_{ij}-1)|U \cap V_i||U \cap V_j||\\
&= 2|d(U \cap V_i, U \cap V_j) - d_{ij}||U \cap V_i||U \cap V_j| \leq 2\epsilon\left(\frac{n}{k}\right)^2.
\end{aligned}$$

Thus,

$$\left|\sum_R f(U \cap V_i, U \cap V_j) - \sum_R (2d_{ij}-1)U \cap V_i||U \cap V_j||\right| \leq \frac{k^2}{2} 2\epsilon \left(\frac{n}{k}\right)^2 = \epsilon n^2$$

which proves the claim.

CLAIM 23. $f(U) =_{\frac{7\epsilon}{2}} f^*(U)$.

By definition, $f^*(U) = \sum_{[k]^2}(2d_{ij}-1)|U \cap V_i||U \cap V_j|$ and by Claim 22 we have $f(U) =_{\frac{5\epsilon}{2}} \sum_R (2d_{ij}-1)|U \cap V_i||U \cap V_j|$. Similar computations show

$$\left|\sum_{[k]^2}(2d_{ij}-1)|U \cap V_i||U \cap V_j| - \sum_R (2d_{ij}-1)|U \cap V_i|||U \cap V_j||\right|$$

$$\leq \sum_{I \cup S} |(2d_{ij}-1)||U \cap V_i||U \cap V_j| \leq \binom{k}{2}\left(\epsilon\left(\frac{n}{k}\right)^2 + \epsilon\left(\frac{n}{k}\right)^2\right) \leq \epsilon n^2. \qquad \square$$

From Lemma 18 we can easily conclude that the set S found by the algorithm has discrepancy $|f(S)| \geq (1-\rho)OPT(f)$. Indeed, let T be such that $|f(T)| = OPT(f)$ and S be the set chosen by the algorithm. From the note after Fact 17 we know that $|f^*(S)| \geq |f^*(T)|$ and Lemma 18 implies

$$|f(S) - f^*(S)| \leq \frac{7}{2}\epsilon n^2, |f(T) - f^*(T)| \leq \frac{7}{2}\epsilon n^2.$$

Thus,

$$\begin{aligned}
|f(S)| &= |f^*(S) + f(S) - f^*(S)| \geq |f^*(S)| - |f(S) - f^*(S)| \geq |f^*(T)| - \frac{7}{2}\epsilon n^2\\
&= |f(T) + f^*(T) - f(T)| - \frac{7}{2}\epsilon n^2 \geq |f(T)| - |f^*(T) - f(T)| - \frac{7}{2}\epsilon n^2\\
&\geq |f(T)| - 7\epsilon n^2.
\end{aligned}$$

Since $|f(T)| \geq cn^2$ and $\epsilon = \frac{\rho c}{7}$ we get $|f(S)| \geq (1-\rho)|f(T)|$. □

We will now turn our attention to the linear ordering problem. Let $T_n = (V, A)$ be a tournament with $V = [n]$. We denote by $OPT(T_n) = \max_{P_n} |T_n \cap P_n|$, where max is taken over all transitive tournaments of order n. For a pair of subsets (V_1, V_2) with $V_1 \cap V_2 = \emptyset$ we define the tournament density $d_T(V_1, V_2)$ as follows: $d_T(V_1, V_2) = \frac{arcs(V_1,V_2)}{|V_1||V_2|}$, where $arcs(V_1, V_2)$ is the number of arcs that start at V_1 and end at V_2. Note that $d_T(V_1, V_2) = 1 - d_T(V_2, V_1)$.

Proof of Theorem 4. The ranking σ' can be constructed by the following procedure: Let $\epsilon = (\frac{\rho}{12})^2$.

1. Define an auxiliary graph G_T as $G_T = (V, E)$, where $E = \{\{v_i, v_j\} : i < j, (v_i, v_j) \in A\}$. Let $l = \frac{1}{\epsilon}$ and let $U_i = \{v_{\frac{n}{l}(i-1)}, \dots, v_{\frac{n}{l}i}\}$ where $i = 1, \dots, l$.
2. Apply Lemma 15 to obtain an ϵ-regular partition of V into $V_1 \cup \dots \cup V_k$, which is a refinement of $U_1 \cup \dots \cup U_l$.
3. Check all $k!$ permutations of the sets $\{V_1, \dots, V_k\}$ to find a permutation σ that maximizes $\sum_{1 \leq i_1 < i_2 \leq k} d_T(V_{\sigma(i_1)}, V_{\sigma(i_2)})|V_{\sigma(i_1)}||V_{\sigma(i_2)}|$.
4. Extend σ inside each of V_i in an arbitrary way to obtain the ranking σ' of V.

Let us first observe that in the first two steps of the algorithm we actually construct an ϵ-regular partition of the tournament T, where the regularity is defined as follows.

DEFINITION 24. *A pair of subsets (V_1, V_2) of V with $V_1 \cap V_2 = \emptyset$ is ϵ-regular in tournament (V, A) if for every $W_1 \subset V_1$ with $|W_1| \geq \epsilon|V_1|$, and every $W_2 \subset V_2$ with $|W_2| \geq \epsilon|V_2|$,*

$$|d_T(W_1, W_2) - d_T(V_1, V_2)| \leq \epsilon.$$

Then, since $\max U_i < \min U_j$ for $i < j$, the following fact holds.

FACT 25. *For $i < j$ let $V_i \subset U_i$ and $V_j \subset U_j$. If (V_i, V_j) is ϵ-regular in the graph G_T with density d_{ij}, then the pair (V_i, V_j) is ϵ-regular in tournament T with density $d_T(V_i, V_j) = d_{ij}$.*

Thus $V_1 \cup \dots \cup V_k$ is an ϵ-regular partition of a tournament T. Without loss of generality, we may assume that the optimal ordering of V is $1 < 2 < \dots < n$. For a subset $S \subset V$, define $h(S)$ as the number of arcs of T that agree with the optimal ordering, i.e., $h(S) = |\{(i,j) \in A : i < j, \text{ and } i, j \in S\}|$. For sets $S_1, S_2 \subset V$ with $S_1 \cap S_2 = \emptyset$ let $h(S_1, S_2)$ be the number of arcs of T between S_1 and S_2 that agree with the optimal ordering, i.e., $h(S_1, S_2) = |\{(i,j) \in A : i < j, i \in S_1, j \in S_2 \text{ or } i \in S_2, j \in S_1\}|$. Note that $h(S_1, S_2) = h(S_2, S_1)$. Define sets $Z_j = \{\frac{n}{s}(j-1), \dots, \frac{n}{s}j\}$, where $s = \frac{1}{\sqrt{\epsilon}}$ and $i = 1, \dots, s$. Simple computations show the following.

FACT 26.
1. $\sum_{j=1}^{s} h(Z_j) \leq \frac{\sqrt{\epsilon}}{2} n^2$;
2. $\sum_{i=1}^{k} h(V_i) \leq \frac{\epsilon}{2} n^2$.

Let $W_{ij} = V_i \cap Z_j$ where $i = 1, \dots, k$ and $j = 1, \dots, s$. We define

$$h^* = \sum_{1 \leq j_1 < j_2 \leq s} \sum_{i_1 \neq i_2} d_T(V_{i_1}, V_{i_2})|W_{i_1 j_1}||W_{i_2 j_2}|.$$

We will show that the number of arcs that agree with the optimal ordering cannot be much larger than h^*, namely, the following.

LEMMA 27. $h(V) \leq h^* + \frac{1}{2}(3\sqrt{\epsilon} + 5\epsilon)n^2$.

Before giving a proof we will establish some auxiliary facts.

CLAIM 28. $h(V) \leq \sum_{1 \leq j_1 < j_2 \leq s} \sum_{i_1 \neq i_2} h(W_{i_1 j_1}, W_{i_2 j_2}) + \frac{1}{2}(\epsilon + \sqrt{\epsilon})n^2$.

Indeed, since $\{W_{ij}\}$ is a partition of V we have

$$h(V) \leq \sum_{1\leq j_1<j_2\leq s}\sum_{i_1\neq i_2} h(W_{i_1j_1}, W_{i_2j_2}) + \sum_{i=1}^{k} h(V_i) + \sum_{j=1}^{s} h(Z_j)$$

$$\leq \sum_{1\leq j_1<j_2\leq s}\sum_{i_1\neq i_2} h(W_{i_1j_1}, W_{i_2j_2}) + \frac{\epsilon}{2}n^2 + \frac{\sqrt{\epsilon}}{2}n^2$$

by Fact 26.

We adopt the notation from the proof of Lemma 18. Let $[k]\times[k] = R\cup I$ where $(i_1,i_2)\in I$ if and only if (V_{i_1}, V_{i_2}) is not ϵ-regular in a tournament T. Note that if $(i_1,i_2)\in I$, then either

- $V_{i_1}, V_{i_2} \subset U_i$ for some $i\in[l]$ or
- (V_{i_1}, V_{i_2}) is irregular in the graph G_T.

CLAIM 29. $h(V) \leq \sum_{1\leq j_1<j_2\leq s}\sum_{(i_1,i_2)\in R} h(W_{i_1j_1}, W_{i_2j_2}) + \frac{1}{2}(3\epsilon+\sqrt{\epsilon})n^2$.

To prove the claim we bound $\sum_{1\leq j_1<j_2\leq s}\sum_{(i_1,i_2)\in I} h(W_{i_1j_1}, W_{i_2j_2})$ from above.

$$\sum_{1\leq j_1<j_2\leq s}\sum_{(i_1,i_2)\in I} h(W_{i_1j_1}, W_{i_2j_2}) = \sum_{(i_1,i_2)\in I}\sum_{j_1<j_2} h(W_{i_1j_1}, W_{i_2j_2})$$

$$\leq \sum_{i}^{l} h(U_i) + \epsilon\binom{k}{2}\frac{n^2}{k^2} \leq l\binom{\frac{n}{l}}{2} + \frac{\epsilon}{2}n^2 \leq \epsilon n^2.$$

Thus,

$$h(V) \leq \sum_{1\leq j_1<j_2\leq s}\sum_{(i_1,i_2)\in R} h(W_{i_1j_1}, W_{i_2j_2}) + \frac{1}{2}(3\epsilon+\sqrt{\epsilon})n^2.$$

Finally, let $[s]\times[k] = B\cup S$, where $S = \{(j,i), |W_{ij}| < \epsilon|V_i|\}$.

CLAIM 30. $h(V) \leq \sum_{1\leq j_1<j_2\leq s}\sum\{h(W_{i_1j_1}, W_{i_2j_2}), (i_1,i_2)\in R, (j_1,i_1),(j_2,i_2)\in B\} + \frac{1}{2}(3\epsilon+3\sqrt{\epsilon})n^2$.

Indeed, for $(j_1,i_1)\in S$ we have $h(W_{i_1j_1}, W_{i_2j_2}) < \epsilon|V_{i_1}||W_{i_2j_2}|$. Therefore,

$$\sum_{j_1<j_2}\sum\{h(W_{i_1j_1}, W_{i_2j_2}), (j_1,i_1)\in S, or (j_2,i_2)\in S\} \leq \sum_{[k]\times[k], i_1\neq i_2}\sum_{j_1<j_2}\epsilon|V_{i_1}||W_{i_2j_2}|$$

$$\leq \sum_{[k]\times[k], i_1\neq i_2}\sum_{j_1=1}^{s}\sum_{j_2=1}^{s}\epsilon|V_{i_1}||W_{i_2j_2}| \leq \epsilon s\sum_{[k]\times[k], i_1\neq i_2}|V_{i_1}||V_{i_2}| \leq \epsilon s k^2\frac{n^2}{k^2} = \sqrt{\epsilon}n^2$$

as $s = \frac{1}{\sqrt{\epsilon}}$.

Proof of Lemma 27. To show Lemma 27, we need to prove that $h(V) \leq h^* + \frac{1}{2}(7\epsilon+3\sqrt{\epsilon})n^2$. For $j_1<j_2$ we have

$$h(W_{i_1j_1}, W_{i_2j_2}) = arcs(W_{i_1j_1}, W_{i_2j_2}) = d_T(W_{i_1j_1}, W_{i_2j_2})|W_{i_1j_1}||W_{i_2j_2}|.$$

Since $|W_{i_1j_1}| \geq \epsilon|V_{i_1}|$, $|W_{i_2j_2}| \geq \epsilon|V_{i_2}|$, and (V_{i_1}, V_{i_2}) is ϵ-regular we can approximate $d_T(W_{i_1j_1}, W_{i_2j_2}) \leq d_T(V_{i_1}, V_{i_2}) + \epsilon$. Clearly,

$$\sum_{j_1<j_2}\sum_{i_1\neq i_2} \epsilon|W_{i_1j_1}||W_{i_2j_2}| = \epsilon\sum_{i_1\neq i_2}\sum_{j_1<j_2}|W_{i_1j_1}||W_{i_2j_2}| = \epsilon\sum_{i_1\neq i_2}|V_{i_1}||V_{i_2}| \leq \epsilon n^2.$$

From Claim 30

$$h(V) \leq \sum_{1\leq j_1<j_2\leq s}\sum\{h(W_{i_1j_1}, W_{i_2j_2}), (i_1,i_2)\in R, (j_1,i_1),(j_2,i_2)\in B\} + \frac{1}{2}(3\epsilon+3\sqrt{\epsilon})n^2$$

$$\leq \sum_{j_1<j_2}\sum_{i_1\neq i_2} d_T(V_{i_1}, V_{i_2})|W_{i_1j_1}||W_{i_2j_2}| + \frac{1}{2}(5\epsilon+3\sqrt{\epsilon})n^2 = h^* + \frac{1}{2}(5\epsilon+3\sqrt{\epsilon})n^2. \qquad \square$$

To complete the proof of Theorem 4, we first introduce an auxiliary digraph K with vertices corresponding to sets W_{ij}, weights on arcs corresponding to approximation of the number of arcs that are consistent with optimal ordering. More formally, let K be a complete k-partite, symmetric digraph with a vertex set $V(K) = \{y_{ij} : i \in [k], j \in [s]\}$ and with weights on arcs defined as follows: $w(y_{i_1j_1}, y_{i_2j_2}) = d_T(V_{i_1}, V_{i_2})|W_{i_1j_1}||W_{i_2j_2}|$ if $i_1 \neq i_2$, and $w(y_{i_1j_1}, y_{i_1j_2}) = 0$. Let $Y_i = \bigcup_{j\in[s]}\{y_{ij}\}$. Vertex $y_{ij} \in Y_i$ corresponds to the set $W_{ij} \subset V_i$ and Y_i corresponds to V_i, $\bigcup_{i\in[k]} y_{ij}$ to Z_j. We define the ordering $\prec$ of $V(K)$ in the following way: $y_{i_1j_1} \prec y_{i_2j_2}$ if and only if either $j_1 < j_2$ or $j_1 = j_2$ and $i_1 < i_2$. Then

$$h^* = \sum_{1\leq j_1<j_2\leq s}\sum_{i_1\neq i_2} w(y_{i_1j_1}, y_{i_2j_2}) \leq \sum_{y_{i_1j_1}\prec y_{i_2j_2}} w(y_{i_1j_1}, y_{i_2j_2}).$$

The final part of the proof is based on the following lemma.

LEMMA 31 (ordering lemma). *There exists a permutation $\sigma : [k] \to [k]$ such that for every ordering $\prec$ of $V(K)$*

$$\sum_{y_{i_1j_1}\prec y_{i_2j_2}} w(y_{i_1j_1}, y_{i_2j_2}) \leq \sum_{1\leq i_1<i_2\leq k}\sum_{j_1,j_2\in[s]} w(y_{\sigma(i_1)j_1}, y_{\sigma(i_2)j_2}).$$

In other words, the sum of weights of the arcs is maximized for an ordering $<$ in which $Y_{i_1} < Y_{i_2} < \cdots < Y_{i_k}$. We postpone the proof of Lemma 31 until the end of this section.

LEMMA 32. $h^* \leq \max_\sigma \sum_{1\leq i_1<i_2\leq k} d_T(V_{\sigma(i_1)}, V_{\sigma(i_2)})|V_{\sigma(i_1)}||V_{\sigma(i_2)}|$

Proof. By the ordering lemma, there exists a permutation $\sigma : [k] \to [k]$ such that

$$h^* \leq \sum_{y_{i_1j_1}\prec y_{i_2j_2}} w(y_{i_1j_1}, y_{i_2j_2}) \leq \sum_{1\leq i_1<i_2\leq k}\sum_{j_1,j_2\in[s]} w(y_{\sigma(i_1)j_1}, y_{\sigma(i_2)j_2})$$

$$= \sum_{1\leq i_1<i_2\leq k}\sum_{j_1,j_2\in[s]} d_T(V_{\sigma(i_1)}, V_{\sigma(i_2)})|W_{i_1j_1}||W_{i_2j_2}|$$

$$= \sum_{1\leq i_1<i_2\leq k} d_T(V_{\sigma(i_1)}, V_{\sigma(i_2)})|V_{\sigma(i_1)}||V_{\sigma(i_2)}|$$

$$\leq \max_{\sigma} \sum_{1\leq i_1<i_2\leq k} d_T(V_{\sigma(i_1)}, V_{\sigma(i_2)})|V_{\sigma(i_1)}||V_{\sigma(i_2)}|. \qquad \square$$

The number of arcs that are consistent with constructed ranking σ' is at least $\max_\sigma \sum_{1\leq i_1<i_2\leq k} d_T(V_{\sigma(i_1)}, V_{\sigma(i_2)})|V_{\sigma(i_1)}||V_{\sigma(i_2)}|$, which by Lemma 27 and Lemma 32 is at least $h(V) - \frac{1}{2}(5\epsilon + 3\sqrt{\epsilon})n^2$. When we combine it with the lower bound $h(V) = OPT(T_n) \geq \frac{1}{4}n^2$ mentioned in the introduction we conclude that the number of arcs that are consistent with constructed ordering is at least $(1-\rho)OPT(T_n)$ since $\rho \geq 10\epsilon + 6\sqrt{\epsilon}$. $\square$

We will now prove the ordering lemma.

Proof of the ordering lemma. To prove the lemma, it is sufficient to show that the sum of weights of arcs is maximized for an ordering in which every Y_i is an interval. Let $\prec$ be an ordering of $V(K)$. We denote by $h(\prec)$ the sum $\sum_{y_{i_1j_1}\prec y_{i_2j_2}} w(y_{i_1j_1}, y_{i_2j_2})$ and for every Y_i, where $i = 1,\ldots,k$, we define a gap-number $g_i = gap_\prec(Y_i)$ as the minimum number of intervals I_{ij} such that $Y_i = \bigcup_{j=1}^{g_i+1} I_{ij}$. Note that the gap-numbers depend on the ordering of $V(K)$.

CLAIM 33. *If $gap_\prec(Y_{i_0}) > 0$ then there exists an ordering $\prec^*$ such that*
1. *$h(\prec) \leq h(\prec^*)$,*
2. *$gap_{\prec^*}(Y_{i_0}) < gap_\prec(Y_{i_0})$, and*
3. *$gap_{\prec^*}(Y_i) = gap_{\prec^*}(Y_i)$ for every $i \neq i_0$.*

Applying the claim to $Y_1, Y_2, \ldots, Y_k$, we construct the ordering in which every $g_i = 0$, i.e., all Y_i are intervals. $\square$

Proof of the claim. Since $gap_\prec(Y_{i_0}) > 0$ there exist two intervals I_{i_01}, I_{i_02} such that $I_{i_01}, I_{i_02} \in Y_{i_0}$ and

$$I_{i_01} < I_{i_1j_1} < I_{i_2j_2} < \cdots < I_{i_tj_t} < I_{i_02}.$$

Without loss of generality we may assume that $d_T(V_{i_0}, V_{i_1})|W_{i_1j_1}| + \cdots + d_T(V_{i_0}, V_{i_t})$ $|W_{i_tj_t}| \geq d_T(V_{i_1}, V_{i_0})|W_{i_1j_1}| + \cdots + d_T(V_{i_t}, V_{i_0})|W_{i_tj_t}|$. Then the sum of the weights of arcs between intervals $I_{i_01} < I_{i_1j_1} < I_{i_2j_2} < \cdots < I_{i_tj_t} < I_{i_02}$ is

$$d_T(V_{i_0}, V_{i_1})|W_{i_1j_1}||I_{i_01}| + \cdots + d_T(V_{i_0}, V_{i_t})|W_{i_tj_t}||I_{i_01}|$$

$$+d_T(V_{i_1}, V_{i_0})|W_{i_1j_1}||I_{i_02}| + \cdots + d_T(V_{i_t}, V_{i_0})|W_{i_tj_t}||I_{i_02}|$$

$$\leq d_T(V_{i_0}, V_{i_1})|W_{i_1j_1}||I_{i_01}| + \cdots + d_T(V_{i_0}, V_{i_t})|W_{i_tj_t}||I_{i_01}|$$

$$+d_T(V_{i_0}, V_{i_1})|W_{i_1j_1}||I_{i_02}| + \cdots + d_T(V_{i_t0}, V_{i_t})|W_{i_tj_t}||I_{i_02}|,$$

which equals the sum of weights of arcs between intervals

$$I_{i_01} < I_{i_02} < I_{i_1j_1} < I_{i_2j_2} < \cdots < I_{i_tj_t}.$$

Therefore, we can reduce the number of gaps of Y_{i_0}. $\square$

4. Conclusions and an open problem. In this paper, we considered the weighted version of discrepancy and tournament ranking problems. In the first part of the paper we generalized the approach from [12] to weighted graphs. In the second part we presented algorithms for both problems which were based on the algorithmic regularity lemma. We want to conclude with the following open problem.

OPEN PROBLEM 1. *For a given n, construct an $m \times n$ matrix $M = [m_{ij}]$ of $+1$'s and -1's with m small which has the following property. For every vector $\overrightarrow{u} \in \{-1,1\}^n$*

$$\frac{1}{m}\sum_{i=1}^{m}\left|\sum_{j=1}^{n} m_{ij}u_j\right| \geq c\sqrt{n}$$

for some constant c.

By probabilistic method one can show the existence of matrix M with $m = n$ and a constant $c = 0.0017$ sufficiently small. Note that Hadamard matrices do not possess the required property, taking $\overrightarrow{u}$ as one of the row vectors of M results in $\sum_{i=1}^{m}|\sum_{j=1}^{n} m_{ij}u_j| = n$.

Let us observe that we can use the solution matrix M to our initial problem of finding a sign vector. Namely, for given $\overrightarrow{v_1}, \ldots, \overrightarrow{v_n} \in \{-1,1\}^n$, there is an $O(n^2 m)$ algorithm that finds $\overrightarrow{X} = (X_1, \ldots, X_n) \in \{-1,1\}^n$ such that

$$||X_1\overrightarrow{v_1} + \cdots + X_n\overrightarrow{v_n}|| \geq cn^{3/2}.$$

Indeed, let $\overrightarrow{v_i} = (v_{i,1}, \ldots, v_{i,n})$ and $\overrightarrow{w_j} = (v_{1,j}, \ldots, v_{n,j})$. We can construct a sign vector in the following way: For every row vector $\overrightarrow{m_i}$ of matrix M we compute $\sum_{j=1}^{n}|\langle\overrightarrow{w_j}, \overrightarrow{m_i}\rangle|$ and we choose $\overrightarrow{m_i}$ such that the sum is the largest.

By the property of the matrix M, we know that for every vector $\overrightarrow{w_j}$, $\sum_{i=1}^{m}|\langle\overrightarrow{w_j}, \overrightarrow{m_i}\rangle| \geq cm\sqrt{n}$ and so $\sum_{j=1}^{n}\sum_{i=1}^{m}|\langle\overrightarrow{w_j}, \overrightarrow{m_i}\rangle| \geq cmn^{3/2}$. This implies that if a vector $\overrightarrow{m} = (m_1, \ldots, m_n)$ is chosen by the algorithm, then $\sum_{j=1}^{n}|\langle\overrightarrow{w_j}, \overrightarrow{m}\rangle| \geq cn^{3/2}$. We verify that

$$||\overrightarrow{v_1}m_1 + \cdots + \overrightarrow{v_n}m_n|| = \sum_{j=1}^{n}|v_{1,j}m_1 + \cdots + v_{n,j}m_n| = \sum_{j=1}^{n}|\langle\overrightarrow{w_j}, \overrightarrow{m}\rangle| \geq cn^{3/2}.$$

In computing the sum $\sum_{j=1}^{n}|\langle\overrightarrow{w_j}, \overrightarrow{m_i}\rangle|$ we add n^2 numbers of size $O(1)$. Note that the same argument can be repeated (resulting in different constant c) if $\overrightarrow{v_1}, \ldots, \overrightarrow{v_n} \in \{-1,1\}^k$, and $k = \Theta(n)$.

As long as m is smaller then $n \lg n$ this will improve the time complexity of results in [12]. A similar question can be asked for the weighted case.

Acknowledgments. We would like to thank referees for helpful comments and suggestions.

REFERENCES

[1] N. ALON, R.A. DUKE, H. LEFMANN, V. RÖDL, AND R. YUSTER, *The algorithmic aspects of the regularity lemma*, J. Algorithms, 16 (1994), pp. 80–109.

[2] A. CZYGRINOW, S. POLJAK, AND V. RÖDL, *On the Linear Ordering Problem*, Emory University Technical Report, Atlanta, GA, 1996.

[3] R.A. DUKE, H. LEFMANN, AND V. RÖDL, *A fast algorithm for computing frequencies in a given graph*, SIAM J. Comput., 24 (1995), pp. 598–620.

[4] P. ERDOS AND J. SPENCER, *Probabilistic Methods in Combinatorics*, Academic Press, New York, 1974.

[5] W. FERNANDEZ DE LA VEGA, *On the maximal cardinality of a consistent set of arcs in a random tournament*, J. Combin. Theory Ser. B, 35 (1983), pp. 328–332.

[6] A. FRIEZE AND R. KANNAN, *The regularity lemma and approximation schemes for dense problems*, in Proc. 37th FOCS, 1996, pp. 12–20.

[7] A. Frieze and R. Kannan, *Quick Approximation to Matrices and Applications*, preliminary manuscript, February 1997.
[8] 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.
[9] M. X. Goemans and D. P. Williamson, .878-*Approximation Algorithm for MAX CUT and MAX 2SAT*, in Proc. 26th STOC, 1994, pp. 422–431.
[10] M. Grötschel, M. Jünger, and G. Reinelt, *A cutting plane algorithm for the linear ordering problem*, Oper. Res., 32 (1984), pp. 1195–1220.
[11] J. K. Lenstra, *Sequencing by Enumerative Methods*, in Mathematical Center Tracts 69, Mathematisch Centrum, Amsterdam.
[12] S. Poljak, V. Rödl, and J. Spencer, *Tournament ranking with expected profit in polynomial time*, SIAM J. Discrete Math., 1 (1988), pp. 372–376.
[13] S. Poljak and D. Turzik, *A polynomial time heuristic for certain subgraph optimization problems with guaranteed lower bound*, Discrete Math., 58 (1986), pp. 99–138.
[14] J. Spencer, *Optimal ranking of tournaments*, Networks, 1 (1971), pp. 135–138.
[15] J. Spencer, *Nonconstructive Methods in Discrete Mathematics*, in Studies in Combinatorics, G. C. Rota, ed., Mathematical Association of America, Washington, D.C., 1978, pp. 142–178.
[16] E. Szemerédi, *Regular partitions of graphs*, in Proc. Colloque Internat. CNRS, J.C. Bermond, ed., Paris, 1978, pp. 399-401.

SIAM J. DISCRETE MATH.
Vol. 12, No. 1, pp. 64–77

OPTIMAL BOUNDS FOR MATCHING ROUTING ON TREES*

LOUXIN ZHANG†

Abstract. The permutation routing problem is studied for trees under the matching model. By introducing a novel and useful (so-called) caterpillar tree partition, we prove that any permutation on an n-node tree (and thus graph) can be routed in $\frac{3}{2}n + O(\log n)$ steps. This answers an open problem of Alon, Chung, and Graham [*SIAM J. Discrete Math.*, 7 (1994), pp. 516–530].

Key words. matching routing, off-line algorithms, trees

AMS subject classifications. 05C, 68M, 68R

PII. S0895480197323159

1. Introduction.

Routing problems on networks arise in different fields such as communications, parallel architectures, and very large scale integration (VLSI) theory and have been extensively studied in recent years (see [9, 10] for a comprehensive survey). In this paper, we study permutation routing under the matching model, which was proposed by Alon, Chung, and Graham [2]. The routing of this type is described as follows. Given a graph $G = (V, E)$ with vertex set V and edge set E. Initially, each vertex v of G is occupied by a "packet" p. To each packet p is associated a destination $\pi(v) \in V$ so that distinct packets have distinct destinations, where the permutation π is usually called a routing assignment. Packets can be moved to different vertices of G under the protocol, which at each step a set S of edges sharing no endpoints of G is selected, and packets at the endpoints of each edge in S are interchanged. The goal of routing is to route all the packets to their destinations in a minimum number of parallel steps. The routing model considered here is interesting as its basic building blocks are very simple. Like hot-potato routing or deflection routing [1, 4], the striking feature of the matching model is that it involves no message queues at each node. In addition, best-known off-line routing algorithms for the hypercube, the linear array, and the mesh can be implemented in this routing model [6, 10, 12]. Such a kind of off-line routing algorithms has also been generalized in various Cayley networks [3].

In their paper, Alon, Chung, and Graham [2] investigated this routing problem for a variety of popular networks including trees, complete (bipartite) graphs, hypercubes, expander graphs, and Cayley graphs. One of their interesting results is that any permutation on a tree with n nodes can be routed in at most $3n$ steps, which was improved to $\frac{13}{5}n$ in [13] and then $2n$ in [5, 7]. They also conjecture that the optimal upper bound for trees is $\frac{3}{2}n$. In [13], Roberts, Symvonis, and Zhang proved that any permutation can be routed in at most $n+o(n)$ steps on an n-node d-ary complete trees in which the root has degree d and any other internal node has degree $d+1$ for some fixed $d > 0$. Furthermore, relation routing under a variant of the matching model

*Received by the editors June 17, 1997; accepted for publication (in revised form) March 5, 1998; published electronically January 29, 1999. This work was done at the University of Waterloo and was partially supported by a CGAT grant. A preliminary version of this paper was presented at the 8th *Annual ACM-SIAM Symposium on Discrete Algorithms*, New Orleans, LA, 1997.

http://www.siam.org/journals/sidma/12-1/32315.html

†BioInformatics Center, Kent Ridge Digital Labs, 21 Heng Mui Keng Terrace, Kent Ridge, Singapore 119613 (lxzhang@krdl.org.sg).

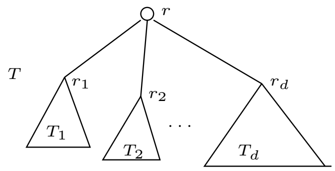


FIG. 1. Partition a tree T.

are also studied by Krizanc and Zhang [8] and Pantziou, Roberts, and Symvonis [11] independently.

This paper favors their conjecture. After a number of refinements are incorporated in Alon, Chung, and Graham's algorithm and their analysis, we obtain a simple $2n$ upper bound for routing on trees in section 2. In section 3, by introducing a novel tree partition and extending our approach, we prove that any permutation can be routed in $\frac{3}{2}n + O(\log n)$ steps on any n-node tree. (As a consequence, any permutation on a n-node graph G can be routed in at most $\frac{3}{2}n + O(\log n)$ steps since G has an n-node spanning subtree.) This bound is quite sharp since the lower bound for the problem is $\frac{3}{2}n$ (see [2]).

2. A simple $2n$-step algorithm. Let T be a tree. For a permutation assignment π on T, we define $rt(\pi, T)$ to be the minimum number of steps required to route π. Further, we define $rt(T)$, the *routing number* of T [2], by $rt(T) = \max_\pi rt(\pi, T)$, where π ranges over all possible permutations on T. We first prove a weak result to demonstrate our approach as a warmup.

THEOREM 2.1. *For any n-node subtree T, $rt(T) \leq 2n$.*

For proving Theorem 2.1, we first introduce some necessary definitions and facts. Given a tree $T = (V, E)$ with vertex set V and edge set E, a class of disjoint subtrees $\{T_i\}_1^s$ is called a *partition* of T if $V(T) = \cup_{i=1}^s V(T_i)$. Let T' be any one of these subtrees. For a routing assignment π, call a packet p that is initially located in T' *improper* if its destination $\pi(p)$ does not belong to T'; call it *proper*, otherwise. T' is *pure* if it does not contain any improper packets, and is *mixed*, otherwise. Since π is a permutation and each node holds one packet, the number of improper packets located in a subtree T' is always equal to the number of improper packets with destinations in T' during routing. This simple observation plays a crucial role in designing off-line algorithms for matching routing.

Proof. We use the standard divide-and-conquer technique to design a $2n$-step off-line algorithm for our purpose, which is a refinement of Alon, Chung, and Graham's algorithm [2, Theorem 1].

For any n-node tree T, there is always a node $r \in V(T)$ that minimizes maximum remaining components. Such a node is called the "centroid." Obviously, each subtree formed by removing the centroid r (and all incident edges) has at most $n/2$ nodes (see Figure 1). Let d subtrees be formed after the removal of r, and let $|T_i| \geq |T_{i+1}|$ for all $i \leq d-1$. Then there exists a nonnegative integer s such that

$$1 + \sum_{j=s+1}^{d} |T_j| \leq |T_1| < 1 + \sum_{j=s}^{d} |T_j|, \qquad (2.1)$$

where we assume $\sum_{j=d+1}^{d} |T_j| = 0$. Let T_r be the subtree (of T) consisting of the node r and subtrees $T_{s+1}, \ldots T_d$. Obviously, $T_1, T_2, \ldots, T_s$ and T_r form a partition of T.

The first objective of our algorithm is to move all improper packets into their destination subtrees in the partition $\{T_r, T_1, T_2, \ldots, T_s\}$. For any subtree T' in the partition, let $I(T')$ and $P(T')$ denote the sets of improper and proper packets in T', respectively.

LEMMA 2.2. *All improper packets, with respect to the partition, can be moved into their destination subtrees in at most*

$$\max\{|P(T_r)|,\ |P(T_j)| \mid j \leq s\} + |I(T_r)| + \sum_{j=1}^{s} |I(T_j)| + s - 1 \leq n + s - 1 \tag{2.2}$$

steps.

Proof. A similar result was proved in [13]. We use a greedy algorithm for moving improper packets into their destination subtrees. In a subtree T', whenever we can interchange an improper and proper packet so as to bring the improper packet close to its destination subtree, we do it; between the vertex r and the roots r_i of subtrees T_i, whenever we can interchange two improper packets so as to bring them into their destination subtrees, we do it.

Our algorithm consists of two phases. In the first phase, we only move improper packets in each subtree T' towards r', the root of T', as many as $m = \max\{|P(T_r)|, |P(T_i)| \mid 1 \leq i \leq s\}$ steps. More specifically, this phase is executed as follows. A node v is said to be in level i of the tree if the distance between it and the root r is i. Let p_v denote the packet at node v before step t. Then packets are moved according to the following rules at step t:

- When $t = 0 \pmod 2$, the packet p_v in even levels is interchanged with an improper packet in a child of v if p_v is proper and the child of v holds an improper packet. When several children hold improper packets, one child is chosen randomly. Otherwise, p_v is not moved at this step.
- When $t = 1 \pmod 2$, the packet p_v in odd levels is interchanged with an improper packet in a child of v if p_v is proper and the child of v holds an improper packet. Otherwise, p_v is not moved at this step.

Since there are at most m proper packets in each subtree T', after this phase is executed, the first improper packet has reached the root r' of T'. Further, in every two steps, a new improper packet will be moved onto r' if there are improper packets. All these facts allow the second phase described below to be executed without delay.

In the second phase, we move improper packets across r into their destination components, as well as keeping on moving up improper packets in each subtree. Before T_r becomes pure, we can move two improper packets into their destination subtrees in every two steps. However, we need extra steps after T_r is pure. After T_r has become pure, the packet p_r at r is proper. If other subtrees still contain improper packets, we have to move p_r into a subtree before moving improper packets cross r and then move it back to r. In order to use the minimum number of steps to move the remaining improper packets, we move the packet p_r to a mixed subtree T' and move p_r back to r when the last improper packet is moved into T'. It can easily be seen that we need one extra step for moving improper packets out of T' for each remaining mixed subtree T'. Since there are at most as many as s such subtrees, the second phase can be finished in at most

$$|I(T_r)| + \sum_{j=1}^{s} |I(T_j)| + s - 1$$

steps, where the subtractive term -1 is derived from the fact that the last two mixed subtrees become pure during the last step (since either at least two mixed subtrees exist or none).

Overall, the whole procedure takes at most

$$\max\{|P(T_r)|,\ |P(T_j)| \mid j \leq s\} + |I(T_r)| + \sum_{j=1}^{s} |I(T_j)| + s - 1 \leq n + s - 1$$

steps. This proves the lemma. □

Now, we continue the proof of Theorem 2.1. After all improper packets have been moved into their destination subtrees, we route each subtree (in parallel).

Obviously, our algorithm is correct. Let $rt(T)$ denote the number of steps needed for routing a permutation on tree T. Then we have

$$rt(T) \leq \max\{rt(T_j), rt(T_r) \mid j \leq s\} + n + s - 1 = rt(T_1) + n + s - 1. \tag{2.3}$$

Let $k = |T_s|$. Then, $k \geq 1$. By inequality (2.1), $|T_1| \leq |T_r| + k - 1$. Since, by the assumption, $|T_j| \geq |T_s|$ for $j \leq s$,

$$s \leq 1 + \frac{n - |T_1| - |T_r|}{k} \leq 1 + \frac{n - 2|T_1| + k - 1}{k}.$$

Therefore, from (2.3) follows that

$$rt(T) \leq \left\lfloor \left(1 + \frac{1}{k}\right) n - \frac{2|T_1|}{k} + rt(T_1) + \frac{k-1}{k} \right\rfloor. \tag{2.4}$$

Further, for a 2-node tree T, $rt(T) = 1 \leq 2 \times 2$. Since $rt(T) \geq |T|$, $k \geq 1$, and $|T_1| \leq \frac{n}{2}$, (2.4) implies that

$$rt(T) \leq 2n.$$

This proves the theorem. □

3. Tight bound for the routing number of trees. In this section, we prove that the routing number of n-node trees is $\frac{3n}{2} + O(\log n)$ by refining our approach used in last section. We first introduce the (so-called) caterpillar partition for trees.

Given a tree T and after the removal of a node v (and all incident edges), T is partitioned into some disjoint subtrees. Such a partition is called a *star partition* of T. Given a positive real number $\gamma < 1$, a star partition is γ-type if each subtree in the partition contains at most $\gamma|T|$ vertices. Recall that every tree has a $\frac{1}{2}$-star partition. However, for any $\gamma < \frac{1}{2}$, a γ-star partition does not always exist for an arbitrary tree.

A partition of T into subtrees is called a *caterpillar partition* if there is a path P in T such that each subtree is rooted at either an nonendpoint v' in P or a child of endpoints of P (see Figure 2). A subtree is *direct* if its root is on P and *indirect* otherwise. The path P is called the *backbone* of the partition. Given two positive real numbers $\gamma, \beta < 1$, a caterpillar partition is called (γ, β)-type if each indirect subtree has at most $\gamma|T|$ nodes and the number of nodes in all direct subtrees is at most $\beta|T|$. Note that by definition, the backbone of a caterpillar partition contains at most $\beta|T| + 2$ nodes.

The proof of our main theorem will use the following result about tree partition. The theorem is of independent interest.

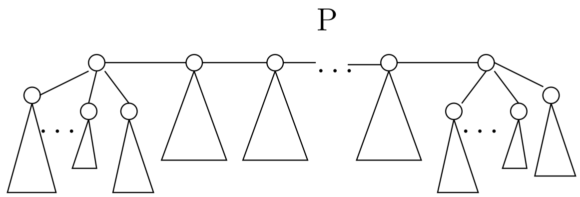


FIG. 2. A caterpillar partition of a tree T. Indirect subtrees are rooted at children of endpoints; direct ones are rooted at internal nodes on the path.

THEOREM 3.1. *Any tree T has either a $\frac{1}{3}$-star or a $(\frac{1}{3}, \frac{1}{3})$-caterpillar partition.*

Proof. Let v be a centroid of an n-node tree T. Let B be the set of all nodes u such that when u is removed there is a component not containing v with at least $n/3$ nodes. If B is empty, then the centroid v induces a $\frac{1}{3}$-star partition. Otherwise, $v \in B$ and B consists of a path in T. This follows from the following facts. First, B induces a subtree. Let $u_1, u_2 \in B$. Then, for any node w on the path between u_1 and u_2 in T, there is a component C not containing v formed after w is removed satisfying the property that C contains u_1 or u_2, and so $w \in B$. Second, if B does not induce a path, then there are three nodes $u_1, u_2, u_3 \in B$ that are adjacent to a common node $u \in B$. In this case, we have three different components not containing v each with at least $n/3$ after nodes u_1, u_2, u_3 are removed—a contradiction. Hence, B consists of a path in T.

It is easy to see that B forms the backbone of a desired $(\frac{1}{3}, \frac{1}{3})$-caterpillar partition. ∎

THEOREM 3.2. *For any n-node tree T,*

$$rt(T) \leq \frac{3n}{2} + 9\log_3 n.$$

Proof. We prove the result by induction. Obviously, the result is trivial for any tree with less than 4 nodes. Thus, we assume that $|T| > 4$ in the rest of our proof.

Let u and v be two leaves in the n-node tree T. They are *sibling* if they have a common parent. Consider a multiset of sibling leaves of T

$$S = \{v_1, v_2, \ldots, v_l \mid l \geq 3, \},$$

where v_i and v_j are different unless $i = 1$ and $j = l$. If the packet $p(v_i)$ at v_i has destination v_{i+1} for all $i \leq l-1$, then we can use l steps to move $l-1$ packets $p(v_i)$, $1 \leq i \leq l-1$, to their destinations. By induction, we can route the rest of the packets to their destinations in $t \leq \frac{3}{2}(n-l+1) + 9\log_3(n-l+1)$. Since $l \geq 3$, we can route π in at most $l + t \leq \frac{3}{2}n + 9\log_3 n$ steps. Therefore, in what follows, we assume that T satisfies the following condition:

($\star\star\star$) There are no sibling leaves v_1, v_2, and v_3 such that the packet $p(v_i)$ has destination v_{i+1} for $i = 1, 2$, where v_1 and v_3 could be same.

By Theorem 3.1, T has either a $\frac{1}{3}$-star or a $(\frac{1}{3}, \frac{1}{3})$-caterpillar partition.

Case 1. T has a $\frac{1}{3}$-star partition.

With $T_1, T_2, \ldots, T_d$ denoting subtrees in such a partition, we assume that the "centroid" is r and $|T_i| \geq |T_{i+1}|$ for $i = 1, 2, \ldots, d-1$ (see Figure 1). We consider the following partition of T:

$$\{T_1, T_2, \ldots, T_d, \{r\}\ \}.$$

We route a permutation by first moving all improper packets into their destination subtrees and then route each subtree recursively as in the proof of Theorem 2.1. In order to reach our desired bound, however, we have to overlap the two phases. For simplicity, we assume that the packet p_r located at node r is proper. (Otherwise, our algorithm takes less steps.) Recall that $P(T_i)$ and $I(T_i)$ denote the sets of proper and improper packets in the subtree T_i for $i \leq d$. After moving all improper packets towards r in $\max_{1 \leq i \leq d} |P(T_i)|$ steps, we start to move improper packets out of subtree T_1 by pushing p_r into T_1. While we move improper packets out of T_1, some improper packets may be moved out of other subtrees. Once T_1 has become pure (and p_r is back to r, of course), we start to route T_1 immediately, as well as move improper packets out of T_2. In general, after moving all improper packets out of T_i, we start to route T_i, as well as move improper packets out of other trees T_j $(j \geq i+1)$, if necessary.

We now analyze the algorithm. Let n_i denote the number of steps taken by the algorithm for moving all improper packets out of T_i after all T_j's $(j < i)$ become pure. Since $|T_i| \leq \frac{1}{3}n$, by induction, all packets with destinations in T_i have been moved to their destinations after at most

$$t_i = \max_j |P_j| + \sum_{j=1}^{i} n_j + rt(T_i) \tag{3.1}$$

steps. Let I_{ij}^k $(i, j \geq k)$ denote the number of improper packets moved from T_i to T_j during the phase of moving improper packets out of T_k. Then, it is not difficult to verify the following formulae:

$$|I(T_i)| = \sum_{l=1}^{i} \sum_{j=l}^{d} I_{ij}^l \tag{3.2}$$

and

$$n_i = 1 + \sum_{j,k \geq i} I_{jk}^i. \tag{3.3}$$

The additive term 1 in the last formula is derived from the fact that we use one extra step to move the packet p_r, located at r, into T_i at the beginning of moving improper packets out of T_i. Plugging formulae (3.2) and (3.3) into (3.1), we have that

$$t_i \leq i + \sum_{j=1}^{d} |T_j| + rt(T_i). \tag{3.4}$$

By the condition $(\star\star\star)$, all subtrees have become pure if all improper packets are moved out of all subtrees that have at least 2 nodes. Thus, we need only to consider those multinode subtrees T'. Since there are at most $\frac{n}{2}$ multinode subtrees, lormula (3.4) can be refined into

$$t_i \leq \min\left\{i, \frac{n}{2}\right\} + \sum_{j=1}^{d} |T_j| + rt(T_i). \tag{3.5}$$

For $i \leq 9$, since $|T_i| \leq \frac{n}{3}$,

$$\min\left\{i, \frac{n}{2}\right\} + |T_i| \leq 9 + \frac{n - |T_i|}{2} \tag{3.6}$$

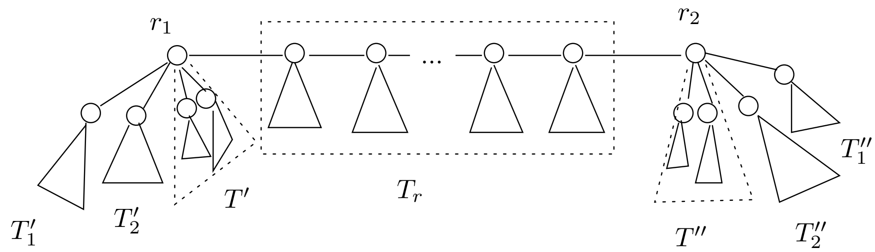


FIG. 3. A $(\frac{1}{3}, \frac{1}{3})$-caterpillar partition of T. Combine all direct subtrees into a middle subtree and some small indirect subtrees at each endpoint into a larger subtree.

holds.

For $9 < i \leq \frac{n}{3}$, we have $|T_i| \leq \frac{n}{i} < \frac{n}{9}$. Thus inequality (3.6) is also true.

For $i > \frac{n}{3}$, we have $|T_i| \leq 2$. Thus inequality (3.6) is still true.

By the induction hypothesis,

$$rt(T_i) \leq \frac{3|T_i|}{2} + 9\log_3 |T_i|.$$

Therefore, (3.5) becomes

$$t_i \leq \frac{3(n - |T_i|)}{2} + 9 + rt(T_i) \leq \frac{3n}{2} + 9\log_3 |T|,$$

which implies that

$$rt(\pi, T) = \max_{1\leq i\leq d} t_i \leq \frac{3n}{2} + 9\log_3 |T|.$$

Case 2. T has a $(\frac{1}{3}, \frac{1}{3})$-caterpillar partition.

Let such a partition $\mathcal{P}$ be illustrated in Figure 3. The backbone P has endpoints r_1 and r_2; indirect trees around the endpoints r_1 and r_2 are T'_i, $1 \leq i \leq m$, and T''_j, $1 \leq j \leq n$, respectively. Without loss of generality, we assume that

$$|T'_i| \geq |T'_{i+1}|,$$

$$|T''_j| \geq |T''_{j+1}|,$$

for all possible i and j. By definition,

$$\sum_{i=1}^{m} |T'_i| + 1 > \frac{1}{3}n \tag{3.7}$$

and

$$\sum_{i=1}^{n} |T''_i| + 1 > \frac{1}{3}n. \tag{3.8}$$

Let T_r denote the union of all directed trees. Set

$$M = \max\{|T_r|, |T'_1|, |T''_1|\}.$$

Since $\mathcal{P}$ is $(\frac{1}{3}, \frac{1}{3})$-caterpillar partition, $M \le \frac{1}{3}n$. Further, by inequality (3.7), there exists c such that

$$\sum_{i=c+1}^{m} |T'_i| + 1 \le M < \sum_{i=c}^{m} |T'_i| + 1,$$

where the sum is 0 if c=m+1. Similarly, there exists d such that

$$\sum_{j=d+1}^{n} |T''_j| + 1 \le M < \sum_{j=d}^{n} |T''_j| + 1.$$

Let T'_{r_1} be the union of $T'_{c+1}, T'_{c+2}, \ldots, T'_m$ and r_1, and let T''_{r_2} be the union of T''_{d+1}, $T''_{d+2}, \ldots, T''_n$ and r_2. Then disjoint subtrees $T_r, T'_{r_1}, T''_{r_2}, T'_i$ $(1 \le i \le c)$, and T''_j $(1 \le j \le d)$ form a partition of T (see Figure 3). For routing permutation π we move all packets into their destination subtrees in the partition and then route each subtree in parallel as in Theorem 2.1. First, we establish the following fact, a generalization of Lemma 2.2 in section 2, the proof of which is somewhat lengthy and involved.

LEMMA 3.3. *Let c' and d' denote the numbers of indirect subtrees T'_i and T''_j satisfying $|T'_i|, |T''_j| \ge 2$, respectively. Then, all improper packets can be moved into their destination subtrees in at most*

$$n + c' + d' + \left\lceil \frac{c-c'}{2} \right\rceil + \left\lceil \frac{d-d'}{2} \right\rceil \tag{3.9}$$

steps.

Proof. Note that $c - c'$ and $d - d'$ are the numbers of indirect, 1-node subtrees around r_1 and r_2 in the partition, respectively. The algorithm for routing improper packets into their destination subtrees consists of two phases.

Let x denote the total number of proper packets with respect to the partition of T, which is described above. Note that an improper packet in each direct subtree has its destination in some indirect subtree around endpoints. In the first phase, improper packets are moved up as many as x steps in each direct or indirect subtree. This phase is detailed in the proof of Theorem 2.1. Roughly speaking, in each direct or indirect subtree T', whenever we can interchange an improper and proper packet so as to bring the improper packet close to the root $r(T')$ of T', we do it.

Recall that after the first phase, in each direct or indirect subtree T' the vertices that improper packets occupy form a subtree T''. In particular, the root of each subtree T' holds an improper packet. Furthermore, if there exist some improper packets out of T'' in T', one improper packet can be moved into T'' in every two steps. This fact guarantees that all improper packets will be moved out of each subtree (and into their destination subtrees) during the second phase.

In the second phase, improper packets will be moved into their destination subtrees. For convenience, we denote all nodes on the backbone P from r_1 to r_2 as

$$v_0(= r_1), v_1, v_2, \ldots, v_k(= r_2).$$

We also assume that k is an odd integer (the case that k is even is similar). Let p_i denote the packet at the node v_i before step t. At step t, we move the packets according to the following rules:

- When $t = 0 \pmod 2$, for every i such that $0 \le i \le \lfloor k/2 \rfloor$, (1) if either p_{2i+1} has destination in a subtree around r_1 and p_{2i} does not, or p_{2i} has destination

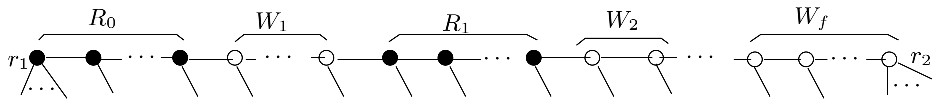


FIG. 4. Packet distribution on the path P. All packets are divided into alternating blocks of red packets and nonred packets, where a red packet is one with destination at some indirect subtree around r_1.

in a subtree around r_2 and p_{2i+1} does not, then interchange the packet p_{2i} and p_{2i+1}. Otherwise, (2) for $i > 0$, interchange p_{2i} with an improper packet in a subtree right below it if p_{2i} is proper. For $i = 0$, either move p_0 into its destination subtree if p_0 is improper, or move p_0 into a chosen subtree around r_1 (to be specified later) if p_0 is proper and T'_{r_1} is pure.

- When $t = 1 \pmod 2$, for every i such that $0 \le i \le \lfloor k/2 \rfloor$, (1) if either p_{2i+2} has destination in a subtree around r_1 and p_{2i+1} does not, or p_{2i+1} has destination in a subtree around r_2 and p_{2i+2} does not, interchange the packet p_{2i+1} and p_{2i+2}. Otherwise, (2) for $i < \lfloor k/2 \rfloor$, interchange p_{2i+1} with an improper packet in a subtree right below it if p_{2i+1} is proper. For $i = \lfloor k/2 \rfloor$, move p_k into its destination subtree if p_k is improper, or move p_k into a chosen subtree around r_2 (to be specified later) if p_k is proper and T''_{r_2} is pure.

Since r_1 is the parent of all subtrees T'_i, $1 \le i \le c$, any improper packet must cross r_1 for being moved into subtrees T'_i. Recall that a subtree is pure if it does not contain any improper packets and is mixed otherwise. After T'_{r_1} is pure, the packet $p(r_1)$ located at root r_1 is proper. If there still exists a mixed subtree T'_i (for some $i \le c$) around r_1, the packet $p(r_1)$ has to be moved into a mixed subtree and moved back later. Such moves of $p(r_1)$ cost extra steps. For reducing extra steps, $p(r_1)$ will be moved into the largest mixed subtree T'_j and back to vertex r_1 during the last step before T'_j becomes pure. Such a choice will reduce the number of extra steps spent at r_1 to c, the number of subtrees around r_1. However, the extra cost can be reduced further if we take care of all 1-node indirect subtrees around r_1.

When all remaining mixed subtrees around r_1 are 1-node subtrees, by our assumption $(\star\star\star)$, an improper packet having destination in these subtrees is still in the middle subtree T_r or in a subtree around r_2. We choose a mixed 1-node subtree into which $p(r_1)$ is moved temporarily according to the arrival time of those improper packets. Suppose the first improper packet of this kind has destination in a 1-node subtree T'_j; we choose any other 1-node, mixed subtree as a temporary destination of $p(r_1)$, unless T'_j is the only one remaining. Such a choice will further reduce the extra steps spent at r_1 from c to $c' + \lceil \frac{c-c'}{2} \rceil$, where c' is the number of multinode subtrees T'_i $(i \le c)$ around r_1. The strategy of reducing extra cost at r_2 is the same.

Now we use the potential method to analyze the complexity of the second phase, which was first used in [2] for studying matching routing. Notice that the second phase will finish once all improper packets with destinations in subtrees around r_1 and r_2 have been moved into their destination subtrees. Therefore, without loss of generality, we may assume that the improper packet that is moved during the last step has destination in a subtree around r_1. For analysis purpose, we call a packet *red* if its destination is in a subtree around r_1 and *blue* if its destination is a subtree around r_2. After each step t, there is a certain distribution of packets on the backbone P. We denote the packet distribution on P in terms of alternating blocks of red and nonred packets as illustrated in Figure 4, where $R_j(t)$ denotes the jth block of red packets and $W_j(t)$ the jth block of nonred packets, which may contain blue improper packets and proper packets, which have destinations in the middle subtree T_r. Each $R_j(t)$ and $W_j(t)$ is nonempty except $W_f(t)$ and $R_0(t)$. Let $I(t)$ denote the set of all improper packets that are off P, and let $g_1(t)$ and $g_2(t)$ denote the numbers of 1-node and multinode, mixed subtrees around r_1 or r_2 after step t, respectively. Finally, define the *potential* $\phi(t)$ after step t by

$$\begin{aligned}\phi(t) = {} & \max(1, |R_0(t)|) - |R_0(t)| + \max(1, |W_{f(t)}(t)|) \\ & -|W_{f(t)}(t)| + \textstyle\sum_{j=1}^{f(t)-1} |W_j(t)| - f(t) + |I(t)| \\ & +g_2(t) + g_1(t)/2,\end{aligned}$$

where $|W_j(t)|$ and $|R_j(t)|$ denote the numbers of packets in blocks $W_j(t)$ and $R_j(t)$, respectively.

We will show that if the last red packet is not moved into its destination subtree, the potential $\phi(t)$ must decrease at the next step. In the rest of our proof, we refer to the packet of a block that is closest to r_1 as the *first* packet and the packet that is closest to r_2 as the *last* packet.

During step $t+1$, the following changes on P may happen:

- Some proper packet p in a nonred block $W_j(t)$, $1 \le j \le f$, is interchanged with an improper packet in the subtree right below p, which has destination in an indirect subtree around an endpoint (i.e., r_1 or r_2).
- The last packet in a nonred block $W_j(t)$ $(1 \le j \le f(t) - 1)$ is interchanged with the first packet of the red block $R_j(t)$.
- In the middle of a nonred block $W_j(t)$ $(1 \le j \le f(t))$, some improper packet p having destination in a subtree around r_2 is interchanged with a proper packet right to p, which has destination in the middle subtree T_r.
- If $W_{f(t)}(t)$ is nonempty, the packet $p(r_2)$ at r_2 can be a proper packet (having destination in T''), an improper packet having destination either in a subtree T''_j around r_2 or in the middle subtree T_r. If $p(r_2)$ is a proper packet and there are still mixed subtrees around r_2, $p(r_2)$ is moved into a chosen mixed subtree; if it is improper and has destination in some T''_j around r_2, it is moved into T''_j; if it is improper and has destination in T_r, then it is interchanged with an improper packet to the left of it, which has destination in a subtree around r_2. Similar events happen at the endpoint r_1.

Based on these observations, we can prove the following facts.

FACT 1. *$|I(t)|$, $g_1(t)$, and $g_2(t)$ is nonincreasing with $t > 0$.*

Proof. The proof follows from definitions. □

FACT 2. *Let $t > 0$. For any nonempty blocks $W_i(t)$ and $R_i(t)$, $1 \le i \le f(t)$, one of the following occurs during step $t+1$:*

(i) *The last packet of $W_i(t)$ and the first packet of $R_i(t)$ are interchanged, or*

(ii) *The last packet of $W_i(t)$ is moved into the subtree right below it. In this case, it is a proper packet with destination in the middle subtree T_r.*

Proof. The events listed in the fact do not happen for some i, $1 \le i \le f(t)$, only if the last packet of $W_i(t)$ is not compared with either the first packet of $R_i(t)$ or a packet in the subtree right below it. As we will show, this cannot happen when $t > 0$. We prove this fact by induction on t.

The basis case $t = 1$ can easily be verified. Assume that the fact is true for $t' \le t - 1$. For any nonempty and adjacent blocks, $W_i(t)$ and $R_i(t)$, we consider the following two cases.

Case 1. $W_i(t)$ is a single-packet block.

Denote the single packet in $W_i(t)$ by q. Consider three subcases.

(a) q was the last packet of some multipacket block $W_{j'}(t-1)$ after step $t-1$. By the induction hypothesis, q is moved right 1 node and formed the single block $W_i(t)$, or the second last packet was interchanged with the red packet right below it. Thus, during step $t+1$, q will be compared with the red packet following it or a packet right below it. So q either moves right or is interchanged with a packet right below it at step $t+1$.

(b) q was the first packet of a two-packet block $W_{j'}(t-1)$ after step $t-1$. Then one of the edges adjacent to the node holding the last packet of $W_{j'}(t-1)$ was active during step t (by the induction hypothesis). This implies that now it is the turn of one of the edges adjacent to node holding q to be active. Thus, q either moves right or is interchanged with a packet right below it at step $t+1$.

(c) q is a packet in the middle of a multipacket block $W_{j'}(t-1)$ after step $t-1$. For q to form a single block after step t, during step t the nonred packet p left to q in $W_{j'}(t-1)$ must be interchanged with a red packet in the subtree right below it and the nonred packet right to q must be moved right one step or moved into a subtree below it. This implies that after step t, the packet right to q is a red packet and also that one of the edges adjacent to the node holding q from the right side is active during step $t+1$. Thus, q either moves right or is interchanged with a packet right below it at step $t+1$.

Case 2. $W_i(t)$ is a multipacket block. By the induction hypothesis, after step $t-1$, the last packet q of the nonred block $W_i(t)$ was either the second last packet of a nonred block $M_{j'}(t-1)$, or it was a middle packet of the block and the packet right to it was moved right one step or moved off the backbone P. This implies that the edges adjacent to q are active during step $t+1$.

Thus we finish the proof of Fact 2. □

FACT 3. *For any* $t > 0$, $\phi(t+1) < \phi(t)$.

Proof. Since in the middle of the backbone moving an improper packet into the backbone from a direct subtree decreases $|I(t)|$ by 1 and $f(t)$ at most by 1, it will not increase the potential ϕ. Therefore, we may assume that no improper packets are moved out of direct subtrees in the middle of the backbone P in the rest of our proof.

We consider the following cases.

Case 1. $R_0(t)$ is an nonempty block.

Since no improper packets are moved into the middle of the backbone, by Fact 2, previously stated, the last packet of $W_j(t)$ is interchanged with the first packet of $R_j(t)$. These moves do not decrease $f(t)$, the number of alternating blocks. Now consider the endpoints r_1. Since R_0 is nonempty, the first red packet is moved into its destination subtree. Thus, the number of improper packets off P, $|I(t)|$, decreases by 1. If the packet swapped out is still red, then the potential ϕ also decreases by 1. Otherwise, $f(t)$ increases by 1 and the potential ϕ decreases by 2. At the endpoint r_2, if the packet $p(r_2)$ at r_2 does not compare with the packet at v_{k-1} and p_{r_2} is an improper packet having destination in a subtree around r_2, then it is moved into its destination subtree and a packet q_{out} is swapped out. If q_{out} is red and improper, then $f(t)$ increases by 1 and $|I(t)|$ decreases by 1. Thus, the potential ϕ decreases by at least 2. If q_{out} is blue and proper, then its destination is in the subtree T''_{r_2} and thus $|I(t)|$ decreases by 1. So is ϕ. If T''_{r_2} is pure, then the number of mixed, indirect 1-node subtrees around the endpoints, $g_1(t)$, decreases at least by 2 or the number of mixed, indirect multinode subtrees around the endpoints, $g_2(t)$, decreases at least by 1; thus ϕ decreases at least by 1. If q_{out} is nonred and improper, the number of improper packets off P, $|I(t)|$, decreases by 1. So is the potential ϕ.

Case 2. $R_0(t)$ is empty.

Since no improper packets are moved into the backbone, by Fact 2, mentioned above, the last packet of $W_j(t)$ is interchanged with the first packet of $R_j(t)$.

If W_1 is a multipacket block, $f(t)$ does not decrease after step $t+1$. We consider the following subcases.

Subcase 2.1. $W_{f(t)}(t)$ is empty. If $R_{f(t)}(t)$ is a multipacket block, the number of alternating blocks, f, increases at least by 1, which implies that the potential ϕ decreases at least by 1. If $R_{f(t)}(t)$ is a singleton, then $W_{f(t+1)}(t+1) = 1$, which implies that $\max(1, |W_f|) - |W_f|$ decreases by 1. Since f does not decrease after step $t+1$, ϕ decreases by 1.

Subcase 2.2. $W_{f(t)}(t)$ is nonempty. Then the last packet p of the block $W_{f(t)}(t)$ is moved into a subtree around r_2. If p is improper, then it moves into its destination subtree by interchanging with a packet q_{out}. The number of improper packets off P, $|I(t)|$, decreases by 1. If q_{out} is blue or having destination in the middle subtree T_r, then the potential ϕ also decreases by 1. If q_{out} is red and improper, then $f(t)$ increases by 1 and $\max(1, |W_f|) - |W_{f)}|$ increases by 1. Thus, ϕ still decreases by 1. If q_{out} is blue and proper, then its destination is in the subtree T''_{r_2}. If T''_{r_2} is pure, then the number of mixed, indirect 1-node subtrees around the endpoints, $g_1(t)$, decreases at least by 2 or the number of mixed, indirect multinode subtrees around the endpoints, $g_2(t)$, decreases at least by 1; thus ϕ decreases at least by 1. If q_{out} is nonred and improper, then the potential also decreases by 1 because the number of improper packets off P, $|I(t)|$, decreases by 1.

If W_1 is a singleton, then $R_0(t+1)$ is 1. Then the difference $\max(1, |R_0|) - |R_0|$ decreases by 1 after step $t+1$. When $W_{f(t)}$ is empty, we consider two cases.

Subcase 2.3. $B_{f(t)-1}(t)$ is a multipacket block.

Then $f(t)$ increases by at least 1 after step $t+1$, and so the potential ϕ decreases at least by 1 after step $t+1$.

Subcase 2.4. $B_{f(t)-1}(t)$ is a singleton.

Then $W_{f(t+1)}(t+1) = 1$. Thus $\max(1, |W_{f(t+1)}|) - |W_{f(t+1)}|$ is 0, which implies that $\phi(t+1) < \phi(t)$. When $W_{f(t)}$ is nonempty, we consider the following cases. If the last packet at r_2 is improper, then it is moved into its destination subtree and so I decreases 1 after $t+1$. Thus, ϕ decreases at least by 1. If the last packet at r_2 is proper and T''_{r_2} is pure, then some subtrees around r_2 have become pure after t. Thus, the number of mixed, indirect multinode subtrees around the endpoints, $g_2(t)$, decreases at least by 1 or the number of mixed, indirect 1-node subtrees around the endpoints, $g_1(t)$, decreases at least by 2. Thus, the potential ϕ decreases at least by 1 after step $t+1$. This finishes the proof of Fact 3. □

We have proved that the potential decreases with t before the last packet is moved into its destination subtree. By definition, the potential ϕ is at most $m = n - x + c' + d' + \lceil (c-c')/2 \rceil + \lceil (d-d')/2 \rceil$, where x is the number of proper packets contained initially in all subtrees. Thus, the second phase takes at most m steps. Since the first phase takes x steps, our greedy algorithm takes at most $n + c' + d' + \lceil (c-c')/2 \rceil + \lceil (d-d')/2 \rceil$ steps. This completes the proof of Lemma 3.3. □

We now continue to prove our theorem. Let $B = c' + d' + (c-c')/2 + (d-d')/2$. Then B is bounded above by $5 + \frac{n-3M}{2}$.

LEMMA 3.4. $B \le 5 + \frac{n-3M}{2}$.

Proof. Let $k' = |T'_c|$ and $k'' = |T''_d|$. Then

$$M \ge |T'_{r_1}| > M - k'$$

and

$$M \ge |T''_{r_2}| > M - k''.$$

If $M = |T_r|$, then

$$c' + \frac{(c-c')}{2} \le \frac{\sum_{i=1}^{m} |T'_i| - M + k'}{\max\{k', 2\}} + 1$$

and

$$d' + \frac{(d-d')}{2} \le \frac{\sum_{j=1}^{n} |T''_j| - M + k''}{\max\{k'', 2\}} + 1.$$

Since $n = \sum_{i=1}^{m} |T'_i| + \sum_{j=1}^{n} |T''_j| + M$,

$$B \le \frac{n-3M}{2} + 4.$$

If $M = |T'_1|$, then

$$c' + \frac{(c-c')}{2} = \frac{\sum_{i=1}^{m} |T'_i| - 2M + k'}{\max\{k', 2\}} + 2$$

and

$$d' + \frac{(d-d')}{2} \le \frac{\sum_{j=1}^{n} |T''_j| - M + k''}{\max\{k'', 2\}} + 1.$$

Thus, $B \le 5 + \frac{n-3M}{2}$. The case $M = |T''_1|$ is treated symmetrically. This finishes the proof of Lemma 3.4. ∎

Let $rt(M) = \max_{|T| \le M} rt(T)$. Combining Lemma 3.4 and (3.9), we have

$$rt(\pi, T) \le n + 2 + B + rt(M) \le n + 7 + \frac{n-3M}{2} + rt(M).$$

By the induction hypothesis, we have

$$rt(\pi, T) \le \frac{3}{2}n + 9\log_3 n.$$

This completes our proof. ∎

4. Concluding remarks. We have proved the optimal upper bound $\frac{3n}{2} + O(\log n)$ for the routing number of n-node trees. In order to prove the result, the caterpillar partition of trees is introduced. Such a partition is novel and may find some interesting applications in studying open problems remaining for the subject of matching routing (see [2]) and other combinatorial problems related to trees and graphs in general.

Acknowledgments. The author would like to thank J. Buss and M. Li for helpful conversations and A. Symvonis for useful discussions and reading the first draft of this paper. He would also like to thank the referee for helpful suggestions in revising the paper, especially for simplifying the proof of Theorem 3.1.

REFERENCES

[1] P. Baran, *On distributed communications networks*, IEEE Trans. Comm. Systems, 12 (1964), pp. 1–9.
[2] N. Alon, F. R. K. Chung, and R. L. Graham, *Routing permutations on graphs via matchings*, SIAM J. Discrete Math., 7 (1994), pp. 516–530.
[3] M. Baumslag and F. Annexstein, *A unified framework for off-line permutation routing in parallel networks*, Math. Systems Theory, 24 (1991), pp. 233–251.
[4] U. Feige and P. Raghaven, *Exact analysis of hot potato routing*, in Proc. 33rd Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1992, pp. 553–562.
[5] W. Goddard, *private communication*, 1997.
[6] N. Haberman, *Parallel Neighbor-Sort (or the Glory of the Induction Principle)*, Tech. report AD-759 248, National Technical Information Services, U.S. Department of Commerce, Springfield, VA, 1972.
[7] P. Hoyer and K. Larson, *Permutation Routing via Matchings*, Tech. report 96-16, Institut for Matematik og Datalogi, Odense Universitet, Denmark, 1996.
[8] D. Krizanc and L. Zhang, *Many-to-one packet routing via matching*, in Proc. 3rd Annual International Conference on Computing and Combinatorics, Shanghai, 1997, Lecture Notes in Comput. Sci. 1276, Springer-Verlag, New York, pp. 11–17.
[9] T. Leighton, *Methods for message routing in parallel machines*, in Proc. 24th Annual ACM Symposium on Theory of Computing, ACM, New York, 1992, pp. 77–96.
[10] T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan–Kaufmann, San Francisco, CA, 1992.
[11] G. Pantziou, A. Roberts, and A. Symvonis, *Dynamic tree routing under the "matching with consumption" model*, in Proc. 7th International Symposium on Algorithms and Computation, Osaka, 1996, Lecture Notes in Comput. Sci. 1178, Springer-Verlag, New York, pp. 275–284.
[12] M. Ramras, *Routing permutations on a graph*, Networks, 23 (1993), pp. 391–398.
[13] A. Roberts, A. Symvonis, and L. Zhang, *Routing on trees via matchings*, in Proc. 4th Workshop on Algorithms and Data Structures, Kingston, Ontario, Canada, 1995, Lecture Notes in Comput. Sci. 955, Springer-Verlag, New York, pp. 251–263.

SIAM J. DISCRETE MATH.
Vol. 12, No. 1, pp. 78–90

ON-LINE DIFFERENCE MAXIMIZATION*

MING-YANG KAO† AND STEPHEN R. TATE‡

Abstract. In this paper we examine problems motivated by on-line financial problems and stochastic games. In particular, we consider a sequence of entirely arbitrary distinct values arriving in random order, and must devise strategies for selecting low values followed by high values in such a way as to maximize the expected gain in rank from low values to high values.

First, we consider a scenario in which only one low value and one high value may be selected. We give an optimal on-line algorithm for this scenario, and analyze it to show that, surprisingly, the expected gain is $n - O(1)$, and so differs from the best possible *off-line* gain by only a constant additive term (which is, in fact, fairly small—at most 15).

In a second scenario, we allow multiple nonoverlapping low/high selections, where the total gain for our algorithm is the sum of the individual pair gains. We also give an optimal on-line algorithm for this problem, where the expected gain is $n^2/8 - \Theta(n \log n)$. An analysis shows that the optimal expected off-line gain is $n^2/6 + \Theta(1)$, so the performance of our on-line algorithm is within a factor of 3/4 of the best off-line strategy.

Key words. analysis of algorithms, on-line algorithms, financial games, secretary problem

AMS subject classifications. 68Q20, 68Q25

PII. S0895480196307445

1. Introduction.

In this paper, we examine the problem of accepting values from an on-line source and selecting values in such a way as to maximize the difference in the ranks of the selected values. The input values can be arbitrary distinct real numbers, and thus we cannot determine with certainty the actual ranks of any input values until we see all of them. Since we only care about their ranks, an equivalent way of defining the input is as a sequence of n integers $x_1, x_2, \ldots, x_n$, where $1 \leq x_i \leq i$ for all $i \in \{1, \ldots, n\}$, and input x_i denotes the rank of the ith input item among the first i items. These ranks uniquely define an ordering of all n inputs, which can be specified with a sequence of ranks $r_1, r_2, \ldots, r_n$, where these ranks form a permutation of the set $\{1, 2, \ldots, n\}$. We refer to the r_i ranks as *final ranks*, since they represent the rank of each item among the final set of n inputs. We assume that the inputs come from a probabilistic source such that all permutations of n final ranks are equally likely.

The original motivation for this problem came from considering on-line financial problems [2, 4, 7, 8, 9], where maximizing the difference between selected items naturally corresponds to maximizing the difference between the buying and selling prices of an investment. While we use generic terminology in order to generalize the setting (for example, we make a "low selection" rather than pick a "buying price"), many of the problems examined in this paper are easily understood using notions from investing. This paper is a first step in applying on-line algorithmic techniques to realistic on-line investment problems.

While the original motivation comes from financial problems, the current input model has little to do with realistic financial markets, and is selected for its mathe-

*Received by the editors July 29, 1996; accepted for publication (in revised form) March 5, 1998; published electronically January 29, 1999.

http://www.siam.org/journals/sidma/12-1/30744.html

†Department of Computer Science, Yale University, New Haven, CT 06520 (kao-ming-yang@cs.yale.edu). The research of this author was supported in part by NSF grant CCR-9531028.

‡Department of Computer Science, University of North Texas, Denton, TX 76208 (srt@cs.unt.edu). The research of this author was supported in part by NSF grant CCR-9409945.

matical cleanness and its relation to fundamental problems in stochastic games. The main difference between our model and more realistic financial problems is that in usual stock trading, optimizing rank-related quantities is not always correlated to optimizing profits in the dollar amount. However, there are some strong similarities as well, such as exotic financial derivatives based on quantities similar to ranks [20].

The current formulation is closely related to an important mathematical problem known as the *secretary problem* [11, 6], which has become a standard textbook example [3, 5, 19], and has been the basis for many interesting extensions (including [1, 14, 15, 17, 18]). The secretary problem comes from the following scenario: A set of candidates for a single secretarial position are presented in random order. The interviewer sees the candidates one at a time, and must make a decision to hire or not to hire immediately upon seeing each candidate. Once a candidate is passed over, the interviewer may not go back and hire that candidate. The general goal is to maximize either the probability of selecting the top candidate, or the expected rank of the selected candidate. This problem has also been stated with the slightly different story of a princess selecting a suitor [3, p. 110]. More will be made of the relationship between our current problem and the secretary problem in section 2, and for further reading on the secretary problem, we refer the reader to the survey by Freeman [10].

As mentioned above, we assume that the input comes from a random source in which all permutations of final ranks $1, 2, \dots, n$ are equally likely. Thus, each rank x_i is uniformly distributed over the set $\{1, 2, \dots, i\}$, and all ranks are independent of one another. In fact, this closely parallels the most popular algorithm for generating a random permutation [13, p. 139]. A natural question to ask is, knowing the relative rank x_i of the current input, what is the expected final rank of this item (i.e., $E[r_i|x_i]$)? Due to the uniform nature of the input source, the final rank of the ith item simply scales up with the number of items left in the input sequence, and so $E[r_i|x_i] = \frac{n+1}{i+1}x_i$ (a simple proof of this is given in Appendix A).

Since all input ranks x_i are independent and uniformly distributed, little can be inferred about the future inputs. We consider games in which a player watches the stream of inputs, and can select items as they are seen; however, if an item is passed up then it is gone for good and may not be selected later. We are interested in strategies for two such games:

- Single pair selection: In this game, the player should make two selections, the first being the *low selection* and the second being the *high selection*. The goal of the player is to maximize the difference between the final ranks of these two selections. If the player picks the low selection upon seeing input x_ℓ at time step ℓ, and picks the high selection as input x_h at time step h, then the *profit* given to the player at the end of the game is the difference in final ranks of these items: $r_h - r_\ell$.
- Multiple pair selection: In this game, the player makes multiple choices of low/high pairs. At the end of the game the difference in final ranks of each selected pair of items is taken, and the differences for all pairs are added up to produce the player's final profit.

The strategies for these games share a common difficulty: If the player waits too long to make the low selection, he risks not having enough choices for a good high selection; however, making the low selection too early may result in an item selected before any truly low items have been seen. The player in the second game can afford to be less selective. If one chosen pair does not give a large difference, there may still be many other pairs that are good enough to make up for this pair's small difference.

We present optimal solutions to both of the games. For the first game, where the player makes a single low selection and a single high selection, our strategy has expected profit $n - O(1)$. From the derivation of our strategy, it will be clear that the strategy is optimal. Even with full knowledge of the final ranks of all input items, the best expected profit in this game is less than n, and so in standard terms of on-line performance measurement [12, 16], the competitive ratio[1] of our strategy is one. The strength of our on-line strategy is rather intriguing.

For the second game, where multiple low/high pairs are selected, we provide an optimal strategy with expected profit $\frac{1}{8}n^2 - O(n \log n)$. For this problem, the optimal off-line strategy has expected profit of approximately $\frac{1}{6}n^2$, and so the competitive ratio of our strategy is $\frac{4}{3}$.

2. Single low/high selection. This section considers a scenario in which the player may pick a single item as the low selection, and a single later item as the high selection. If the low selection is made at time step ℓ and the high selection is made at time step h, then the expected profit is $E[r_h - r_\ell]$. The player's goal is to use a strategy for picking ℓ and h in order to maximize this expected profit.

As mentioned in the previous section, this problem is closely related to the secretary problem. A great deal of work has been done on the secretary problem and its variations, and this problem has taken a fundamental role in the study of games against a stochastic opponent. Our work extends the secretary problem, and gives complete solutions to two natural variants that have not previously appeared in the literature.

Much insight can be gained by looking at the optimal solution to the secretary problem, so we first sketch that solution below (using terminology from our problem about a "high selection"). To maximize the expected rank of a single high selection, we define the optimal strategy recursively using the following two functions:

- $\mathcal{H}_n(i)$: This is a limit such that the player selects the current item if $x_i \geq \mathcal{H}_n(i)$.
- $R_n(i)$: This is the expected final rank of the high selection if the optimal strategy is followed starting at the ith time step.

Since all permutations of the final ranks are equally likely, if the ith input item has rank x_i among the first i data items, then its expected final rank is $\frac{n+1}{i+1}x_i$. Thus, an optimal strategy for the secretary problem is to select the ith input item if and only if its expected final rank is better than could be obtained by passing over this item and using the optimal strategy from step $i+1$ on. In other words, select the item at time step $i < n$ if and only if

$$\frac{n+1}{i+1}x_i \geq R_n(i+1).$$

If we have not made a selection before the nth step, then we must select the last item, whose rank is uniformly distributed over the range of integers from 1 to n—and so the expected final rank in that case is $R_n(n) = \frac{n+1}{2}$. For $i < n$ we can also define

$$\mathcal{H}_n(i) = \left\lceil \frac{i+1}{n+1} R_n(i+1) \right\rceil,$$

[1] "Competitive ratio" usually refers to the worst-case ratio of on-line to off-line cost; however, in our case inputs are entirely probabilistic, so our "competitive ratio" refers to expected on-line to expected off-line cost—a worst-case measure doesn't even make sense here.

and to force selection at the last time step define $\mathcal{H}_n(n) = 0$. Furthermore, given this definition for $\mathcal{H}_n(i)$, the optimal strategy at step i depends only on the rank of the current item (which is uniformly distributed over the range $1, \ldots, i$) and the optimal strategy at time $i+1$. This allows us to recursively define $R_n(i)$ as follows when $i < n$:

$$\begin{aligned} R_n(i) &= \frac{\mathcal{H}_n(i)-1}{i} R_n(i+1) + \sum_{j=\mathcal{H}_n(i)}^{i} \frac{1}{i} \cdot \frac{n+1}{i+1} j \\ &= \frac{\mathcal{H}_n(i)-1}{i} R_n(i+1) + \frac{n+1}{i(i+1)} \cdot \frac{(i+\mathcal{H}_n(i))(i-\mathcal{H}_n(i)+1)}{2} \\ &= \frac{\mathcal{H}_n(i)-1}{i} \left(R_n(i+1) - \frac{n+1}{2(i+1)} \mathcal{H}_n(i) \right) + \frac{n+1}{2}. \end{aligned}$$

Since $\mathcal{H}_n(n) = 0$ and $R_n(n) = \frac{n+1}{2}$, we have a full recursive specification of both the optimal strategy and the performance of the optimal strategy. The performance of the optimal strategy, taken from the beginning, is $R_n(1)$. This value can be computed by the recursive equations, and was proved by Chow et al. to tend to $n+1-c$, for $c \approx 3.8695$, as $n \to \infty$ [6]. Furthermore, the performance approaches this limit from above, so for all n we have performance greater than $n - 2.87$.

For single pair selection, once a low selection is made we want to maximize the expected final rank of the high selection. If we made the low selection at step i, then we can optimally make the high selection by following the above strategy for the secretary problem, which results in an expected high selection rank of $R_n(i+1)$. How do we make the low selection? We can do this optimally by extending the recursive definitions given above with two new functions:

$\mathcal{L}_n(i)$: This is a limit such that the player selects the current item if $x_i \leq \mathcal{L}_n(i)$.

$P_n(i)$: This is the expected high-low difference if the optimal strategy for making the low and high selections is followed starting at step i.

Thus, if we choose the ith input as the low selection, the expected profit is $R_n(i+1) - \frac{n+1}{i+1} x_i$. We should select this item if that expected profit is no less than the expected profit if we skip this item. This leads to the definition of $\mathcal{L}_n(i)$:

$$\mathcal{L}_n(i) = \begin{cases} 0 & \text{if } i = n, \\ \left\lfloor \frac{i+1}{n+1} \left(R_n(i+1) - P_n(i+1) \right) \right\rfloor & \text{if } i < n. \end{cases}$$

Using $\mathcal{L}_n(i)$, we derive the following profit function:

$$P_n(i) = \begin{cases} 0 & \text{if } i = n, \\ P_n(i+1) + \frac{\mathcal{L}_n(i)}{i} \left(R_n(i+1) - P_n(i+1) - \frac{n+1}{i+1} \cdot \frac{\mathcal{L}_n(i)+1}{2} \right) & \text{if } i < n. \end{cases}$$

From the derivation, it is clear that this is the optimal strategy, and can be implemented by using the recursive formulas to compute the $\mathcal{L}_n(i)$ values. The expected profit of our algorithm is given by $P_n(1)$, which is bounded in the following theorem.

THEOREM 2.1. *Our on-line algorithm for single low/high selection is optimal and has expected profit $n - O(1)$.*

Proof. It suffices to prove that a certain inferior algorithm has expected profit $n - O(1)$. The inferior algorithm is as follows: Use the solution to the secretary problem to select, from the first $\lfloor n/2 \rfloor$ input items, an item with the minimum expected final rank. Similarly, pick an item with maximum expected rank from the second $\lceil n/2 \rceil$ inputs. For simplicity, we initially assume that n is even; see comments at the end of the proof for odd n. Let ℓ be the time step in which the low selection is made, and h the time step in which the high selection is made. Using the bounds from Chow et al. [6], we can bound the expected profit of this inferior algorithm by

$$E[r_h - r_\ell] = E[r_h] - E[r_l] \geq \frac{n+1}{n/2+1}(n/2+1-c) - \frac{n+1}{n/2+1}c$$
$$= \frac{n+1}{n+2}(n+2-4c) = n+1-4c+\frac{4c}{n+2} \ .$$

Chow et al. [6] show that $c \leq 3.87$, and so the expected profit of the inferior algorithm is at least $n - 14.48$. For odd n, the derivation is almost identical, with only a change in the least significant term; specifically, the expected profit of the inferior algorithm for odd n is $n + 1 - 4c + \frac{4c}{n+3}$, which again is at least $n - 14.48$. ∎

3. Multiple low/high selection. This section considers a scenario in which the player again selects a low item followed by a high item, but may repeat this process as often as desired. If the player makes k low and high selections at time steps $\ell_1, \ell_2, \ldots, \ell_k$ and $h_1, h_2, \ldots, h_k$, respectively, then we require that

$$1 \leq \ell_1 < h_1 < \ell_2 < h_2 < \cdots < \ell_k < h_k \leq n.$$

The expected profit resulting from these selections is

$$E[r_{h_1} - r_{\ell_1}] + E[r_{h_2} - r_{\ell_2}] + \cdots + E[r_{h_k} - r_{\ell_k}].$$

3.1. Off-line analysis. Let *interval* j refer to the time period between the instant of input item j arriving and the instant of input item $j+1$ arriving. For a particular sequence of low and high selections, we call interval j *active* if $\ell_i \leq j < h_i$ for some index i. We then amortize the total profit of a particular algorithm B by defining the amortized profit $A_B(j)$ for interval j to be

$$A_B(j) = \begin{cases} r_{j+1} - r_j & \text{if interval } j \text{ is active,} \\ 0 & \text{otherwise.} \end{cases}$$

Note that for a fixed sequence of low/high selections, the sum of all amortized profits is exactly the total profit, i.e.,

$$\sum_{j=1}^{n} A_B(j) = \sum_{j=\ell_1}^{h_1-1}(r_{j+1} - r_j) + \sum_{j=\ell_2}^{h_2-1}(r_{j+1} - r_j) + \cdots + \sum_{j=\ell_k}^{h_k-1}(r_{j+1} - r_j)$$
$$= (r_{h_1} - r_{\ell_1}) + (r_{h_2} - r_{\ell_2}) + \cdots + (r_{h_k} - r_{\ell_k}).$$

For an off-line algorithm to maximize the total profit we need to maximize the amortized profit, which is done for a particular sequence of r_i's by making interval j active if and only if $r_{j+1} > r_j$. Translating this back to the original problem of making low and high selections, this is equivalent to identifying all maximal-length increasing intervals and selecting the beginning and ending points of these intervals as low and high selections, respectively. These observations and some analysis give the following lemma.

LEMMA 3.1. *The optimal off-line algorithm just described has expected profit* $\frac{1}{6}\left(n^2-1\right)$.

Proof. This analysis is performed by examining the expected amortized profits for individual intervals. In particular, for any interval j,

$$\begin{aligned}
E[A_{OFF}(j)] &= Pr[r_{j+1} > r_j] \cdot E[A_j | r_{j+1} > r_j] + Pr[r_{j+1} < r_j] \cdot E[A_j | r_{j+1} < r_j] \\
&= \frac{1}{2} \cdot E[r_{j+1} - r_j | r_{j+1} > r_j] + \frac{1}{2} \cdot 0 \\
&= \frac{1}{2} \sum_{i=1}^{n-1} \sum_{k=i+1}^{n} \frac{Pr[r_{j+1} = k \text{ and } r_j = i]}{Pr[r_{j+1} > r_j]} \cdot (k-i) \\
&= \frac{1}{2} \sum_{i=1}^{n-1} \sum_{k=i+1}^{n} \frac{2}{n(n-1)} (k-i) \\
&= \frac{1}{2} \cdot \frac{2}{n(n-1)} \cdot \frac{(n+1)n(n-1)}{6} \\
&= \frac{n+1}{6}.
\end{aligned}$$

Since there are $n-1$ intervals and the above analysis is independent of the interval number j, summing the amortized profit over all intervals gives the expected profit stated in the lemma. ∎

3.2. On-line analysis. In our on-line algorithm for multiple pair selection, there are two possible states: FREE and HOLDING. In the FREE state, we choose the current item as a low selection if $x_i < \frac{i+1}{2}$; furthermore, if we select an item then we move from the FREE state into the HOLDING state. On the other hand, in the HOLDING state if the current item has $x_i > \frac{i+1}{2}$, then we choose this item as a high selection and move into the FREE state. We name this algorithm OP, which can stand for "opportunistic" since this algorithm makes a low selection whenever the probability is greater than $\frac{1}{2}$ that the next input item will be greater than this one. Later we will see that the name OP could just as well stand for "optimal," since this algorithm is indeed optimal.

The following lemma gives the expected profit of this algorithm. In the proof of this lemma we use the following equality:

$$\sum_{i=1}^{k} \frac{2i}{2i+1} = k + 1 + \frac{1}{2} H_k - H_{2k+1}.$$

LEMMA 3.2. *The expected profit from our on-line algorithm is*

$$E[P_{OP}] = \begin{cases} \dfrac{n+1}{8}\left(n + H_{\frac{n-2}{2}} - 2H_{n-1}\right) & \textit{if } n \textit{ is even,} \\[2ex] \dfrac{n+1}{8}\left(n + H_{\frac{n-1}{2}} - 2H_n + \dfrac{1}{n}\right) & \textit{if } n \textit{ is odd.} \end{cases}$$

In cleaner forms we have $E[P_{OP}] = \frac{n+1}{8}(n - H_n + \Theta(1)) = \frac{1}{8}n^2 - \Theta(n \log n)$.

Proof. Let R_i be the random variable of the final rank of the ith input item. Let $A_{OP}(i)$ be the amortized cost for interval i as defined in section 3.1. Since $A_{OP}(i)$ is nonzero only when interval i is active,

$$\begin{aligned} E[A_{OP}(i)] &= E[A_{OP}(i)|\text{Interval } i \text{ is active}] \cdot Prob[\text{Interval } i \text{ is active}] \\ &= E[R_{i+1} - R_i|\text{Interval } i \text{ is active}] \cdot Prob[\text{Interval } i \text{ is active}]. \end{aligned}$$

Therefore,

$$\begin{aligned} E[P_{OP}] &= \sum_{i=1}^{n-1} E[A_{OP}(i)] \\ &= \sum_{i=1}^{n-1} E[R_{i+1} - R_i|\text{Interval } i \text{ is active}] \cdot Prob[\text{Interval } i \text{ is active}]. \end{aligned}$$

Under what conditions is an interval active? If $x_i < \frac{i+1}{2}$ this interval is certainly active. If the algorithm was not in the HOLDING state prior to this step, it would be after seeing input x_i. Similarly, if $x_i > \frac{i+1}{2}$ the algorithm must be in the FREE state during this interval, and so the interval is not active. Finally, if $x_i = \frac{i+1}{2}$ the state remains what it has been for interval $i-1$. Furthermore, since i must be odd for this case to be possible, $i-1$ is even, and x_{i-1} cannot be $\frac{i}{2}$ (and thus x_{i-1} unambiguously indicates whether interval i is active). In summary, determining whether interval i is active requires looking at only x_i and occasionally x_{i-1}. Since the expected amortized profit of step i depends on whether i is odd or even, we break the analysis up into these two cases below.

Case 1. i is even. Note that $Prob[x_i < \frac{i+1}{2}] = \frac{1}{2}$, and x_i cannot be exactly $\frac{i+1}{2}$, which means that with probability $\frac{1}{2}$ interval i is active. Furthermore, R_{i+1} is independent of whether interval i is active or not, and so

$$\begin{aligned} E[A_{OP}(i)|\text{Interval } i \text{ is active}] &= E[R_{i+1}] - E[R_i|\text{Interval } i \text{ is active}] \\ &= \frac{n+1}{2} - \frac{n+1}{i+1} \sum_{j=1}^{i/2} \frac{2}{i} j \\ &= \frac{n+1}{2} - \frac{n+1}{i+1} \cdot \frac{2}{i} \cdot \frac{i(i+2)}{8} \\ &= \frac{n+1}{4} \cdot \frac{i}{i+1}. \end{aligned}$$

Case 2. i is odd. Since interval 1 cannot be active, we assume that $i \geq 3$. We need to consider the case in which $x_i = \frac{i+1}{2}$, and so

$$\begin{aligned} &Prob[\text{Interval } i \text{ is active}] \\ &= Prob\left[x_i < \frac{i+1}{2}\right] + Prob\left[x_i = \frac{i+1}{2}\right] \cdot Prob\left[x_{i-1} < \frac{i}{2}\right] \\ &= \frac{i-1}{2i} + \frac{1}{i} \cdot \frac{1}{2} = \frac{1}{2}. \end{aligned}$$

Computing the expected amortized cost of interval i is slightly more complex than in Case 1.

$$\begin{aligned} &E[A_{OP}(i)|\text{Interval } i \text{ is active}] \\ &= E[R_{i+1}] - E[R_i|\text{Interval } i \text{ is active}] \end{aligned}$$

$$
\begin{aligned}
&= \frac{n+1}{2} - \frac{n+1}{i+1}\left(\sum_{j=1}^{(i-1)/2} \frac{2}{i}j + \frac{1}{i}\cdot\frac{i+1}{2}\right) \\
&= \frac{n+1}{2} - \frac{n+1}{i+1}\left(\frac{2}{i}\cdot\frac{(i-1)(i+1)}{8} + \frac{1}{i}\cdot\frac{i+1}{2}\right) \\
&= \frac{n+1}{2} - \frac{n+1}{i+1}\cdot\frac{(i+1)(i+1)}{4i} \\
&= \frac{n+1}{4}\cdot\frac{i-1}{i}.
\end{aligned}
$$

Combining both cases,

$$
\begin{aligned}
E[P_{OP}] &= \sum_{i=1}^{n-1} E[A_{OP}(i)|\text{Interval } i \text{ is active}]\cdot Prob[\text{Interval } i \text{ is active}] \\
&= \frac{n+1}{8}\left(\sum_{k=1}^{\lfloor (n-2)/2\rfloor} \frac{2k}{2k+1} + \sum_{k=1}^{\lfloor (n-1)/2\rfloor} \frac{2k}{2k+1}\right),
\end{aligned}
$$

where the first sum accounts for the odd terms of the original sum, and the second sum accounts for the even terms.

When n is even this sum becomes

$$
\begin{aligned}
E[P_{OP}] &= \frac{n+1}{8}\left(\sum_{k=1}^{\lfloor (n-2)/2\rfloor} \frac{2k}{2k+1} + \sum_{k=1}^{\lfloor (n-1)/2\rfloor} \frac{2k}{2k+1}\right) \\
&= \frac{n+1}{8}\left(2\sum_{k=1}^{(n-2)/2} \frac{2k}{2k+1}\right) \\
&= \frac{n+1}{8}\left(n + H_{\frac{n-2}{2}} - 2H_{n-1}\right),
\end{aligned}
$$

which agrees with the claim in the lemma. When n is odd the sum can be simplified as

$$
\begin{aligned}
E[P_{OP}] &= \frac{n+1}{8}\left(\sum_{k=1}^{\lfloor (n-2)/2\rfloor} \frac{2k}{2k+1} + \sum_{k=1}^{\lfloor (n-1)/2\rfloor} \frac{2k}{2k+1}\right) \\
&= \frac{n+1}{8}\left(2\sum_{k=1}^{(n-1)/2} \frac{2k}{2k+1} - \frac{n-1}{n}\right) \\
&= \frac{n+1}{8}\left(n + H_{\frac{n-1}{2}} - 2H_n + \frac{1}{n}\right),
\end{aligned}
$$

which again agrees with the claim in the lemma. The simplified forms follow the fact that for any odd $n \geq 3$ we can bound $\frac{1}{n} \leq H_n - H_{\frac{n-1}{2}} \leq \ln 2 + \frac{1}{n}$. □

Combining this result with that of section 3.1, we see that our on-line algorithm has expected profit 3/4 of what could be obtained with full knowledge of the future. In terms of competitive analysis, our algorithm has competitive ratio 4/3, which means that not knowing the future is not terribly harmful in this problem!

3.3. Optimality of our on-line algorithm. This section proves that algorithm OP is optimal. We will denote permutations by a small Greek letter with a subscript giving the size of the permutation; in other words, a permutation on the set $\{1, 2, \ldots, i\}$ may be denoted ρ_i or σ_i.

A permutation on i items describes fully the first i inputs to our problem, and given such a permutation we can also compute the permutation described by the first $i-1$ inputs (or $i-2$, etc.). We will use the notation $\sigma_i|_{i-1}$ to denote such a restriction. This is not just a restriction of the domain of the permutation to $\{1, \ldots, i-1\}$, since unless $\sigma_i(i) = i$ this simplistic restriction will not form a valid permutation.

Upon seeing the ith input, an algorithm may make one of the following moves: it may make this input a low selection; it may make this input a high selection; or it may simply ignore the input and wait for the next input. Therefore, any algorithm can be entirely described by a function which maps permutations (representing inputs of arbitrary length) into this set of moves. We denote such a move function for algorithm B by M_B, which for any permutation σ_i maps $M_B(\sigma_i)$ to an element of the set {"low" "high" "wait"}. Notice that not all move functions give valid algorithms. For example, it is possible to define a move function that makes two low selections in a row for certain inputs, even though this is not allowed by our problem.

We define a generic HOLDING state just as we did for our algorithm. An algorithm is in the HOLDING state at time i if it has made a low selection, but has not yet made a corresponding high selection. For algorithm B we define the set $L_B(i)$ to be the set of permutations on i items that result in the algorithm being in the HOLDING state after processing these i inputs. We explicitly define these sets using the move function:

$$L_B(i) = \begin{cases} \{\sigma_i | M_B(\sigma_i) = \text{"low"}\} & \text{if } i = 1, \\ \{\sigma_i | M_B(\sigma_i) = \text{"low" or} \\ \quad (M_B(\sigma_i) = \text{"wait" and } \sigma_i|_{i-1} \in L_B(i-1))\} & \text{if } i > 1. \end{cases}$$

The $L_B(i)$ sets are all we need to compute the expected amortized profit for interval i, since

$$\begin{aligned} E[A_B(i)] &= Prob[\text{Interval } i \text{ is active}] \cdot E[R_{i+1} - R_i | \text{Interval } i \text{ is active}] \\ &= \frac{|L_B(i)|}{i!} \left(\frac{n+1}{2} - \frac{n+1}{i+1} \sum_{\rho_i \in L_B(i)} \frac{1}{|L_B(i)|} \rho_i(i) \right) \\ &= \frac{n+1}{i!} \left(\frac{|L_B(i)|}{2} - \frac{1}{i+1} \sum_{\rho_i \in L_B(i)} \rho_i(i) \right). \end{aligned}$$

We use the above notation and observations to prove the optimality of algorithm OP.

THEOREM 3.3. *Algorithm OP is an optimal algorithm for the multiple pair selection problem.*

Proof. Since the move functions (which define specific algorithms) work on permutations, we will fix an ordering of permutations in order to compare strategies. We order permutations first by their size, and then by a lexicographic ordering of the actual permutations. When comparing two different algorithms B and C, we start enumerating permutations in this order and count how many permutations cause the same move in B and C, stopping at the first permutation σ_i for which $M_B(\sigma_i) \neq M_C(\sigma_i)$, i.e., the first permutation for which the algorithms make different moves. We call the number of permutations that produce identical moves in this comparison process the *length of agreement between B and C*.

To prove the optimality of our algorithm by contradiction, we assume that it is not optimal, and of all the optimal algorithms let B be the algorithm with the longest possible length of agreement with our algorithm OP. Let σ_k be the first permutation in which $M_B(\sigma_k) \neq M_{OP}(\sigma_k)$. Since B is different from OP at this point, at least one of the following cases must hold:

(a) $\sigma_k|_{k-1} \notin L_B(k-1)$ and $\sigma_k(k) < \frac{k+1}{2}$ and $M_B(\sigma_k) \neq$ "low" (i.e., algorithm B is not in the HOLDING state, gets a low rank input, but does not make it a low selection).

(b) $\sigma_k|_{k-1} \notin L_B(k-1)$ and $\sigma_k(k) \geq \frac{k+1}{2}$ and $M_B(\sigma_k) \neq$ "wait" (i.e., algorithm B is not in the HOLDING state, gets a high rank input, but makes it a low selection anyway).

(c) $\sigma_k|_{k-1} \in L_B(k-1)$ and $\sigma_k(k) > \frac{k+1}{2}$ and $M_B(\sigma_k) \neq$ "high" (i.e., algorithm B is in the HOLDING state, gets a high rank input, but doesn't make it a high selection).

(d) $\sigma_k|_{k-1} \in L_B(k-1)$ and $\sigma_k(k) \leq \frac{k+1}{2}$ and $M_B(\sigma_k) \neq$ "wait" (i.e., algorithm B is in the HOLDING state, gets a low rank input, but makes it a high selection anyway).

In each case, we will show how to transform algorithm B into a new algorithm C such that C performs at least as well as B, and the length of agreement between C and OP is longer than that between B and OP. This provides the contradiction that we need.

Case (a). Algorithm C's move function is identical to B's except for the following values:

$$M_C(\sigma_k) = \text{“low”},$$

$$M_C(\rho_{k+1}) = \begin{cases} \text{“high”} & \text{if } \rho_{k+1}|_k = \sigma_k \text{ and } M_B(\sigma_{k+1}) = \text{“wait”}, \\ \text{“wait”} & \text{if } \rho_{k+1}|_k = \sigma_k \text{ and } M_B(\sigma_{k+1}) = \text{“low”}, \\ M_B(\rho_{k+1}) & \text{otherwise.} \end{cases}$$

In other words, algorithm C is the same as algorithm B except that we "correct B's error" of not having made this input a low selection. The changes of the moves on input $k+1$ insures that $L_C(k+1)$ is the same as $L_B(k+1)$. It is easily verified that the new sets $L_C(i)$ (corresponding to the HOLDING state) are identical to the sets $L_B(i)$ for all $i \neq k$. The only difference at k is the insertion of σ_k, i.e., $L_C(k) = L_B(k) \cup \{\sigma_k\}$.

Let P_B and P_C be the profits of B and C, respectively. Since their amortized costs differ only at interval k,

$$\begin{aligned} &E[P_C - P_B] \\ &= E[A_C(k)] - E[A_B(k)] \\ &= \frac{n+1}{k!}\left(\frac{|L_C(k)|}{2} - \frac{1}{k+1}\sum_{\rho_k \in L_C(k)} \rho_k(k)\right) \\ &\quad -\frac{n+1}{k!}\left(\frac{|L_B(k)|}{2} - \frac{1}{k+1}\sum_{\rho_k \in L_B(k)} \rho_k(k)\right) \\ &= \frac{n+1}{k!}\left(\frac{1}{2} - \frac{1}{k+1}\sigma_k(k)\right). \end{aligned}$$

By one of the conditions of Case (a), $\sigma_k(k) < \frac{k+1}{2}$, so we finish this derivation by noting that

$$E[P_C - P_B] = \frac{n+1}{k!}\left(\frac{1}{2} - \frac{1}{k+1}\sigma_k(k)\right) > \frac{n+1}{k!}\left(\frac{1}{2} - \frac{1}{k+1}\cdot\frac{k+1}{2}\right) = 0.$$

Therefore, the expected profit of algorithm C is greater than that of B.

Case (b). As in Case (a) we select a move function for algorithm C that causes only one change in the sets of HOLDING states, having algorithm C not make input k a low selection. In particular, these sets are identical with those of algorithm B with the one exception that $L_C(k) = L_B(k) - \{\sigma_k\}$. Analysis similar to Case (a) shows

$$E[P_C - P_B] = \frac{n+1}{k!}\left(\frac{1}{k+1}\sigma_k(k) - \frac{1}{2}\right) \geq \frac{n+1}{k!}\left(\frac{1}{k+1}\cdot\frac{k+1}{2} - \frac{1}{2}\right) = 0.$$

Case (c). In this case we select a move function for algorithm C such that $L_C(k) = L_B(k) - \{\sigma_k\}$, resulting in algorithm C selecting input k as a high selection, and giving an expected profit gain of

$$E[P_C - P_B] = \frac{n+1}{k!}\left(\frac{1}{k+1}\sigma_k(k) - \frac{1}{2}\right) > \frac{n+1}{k!}\left(\frac{1}{k+1}\cdot\frac{k+1}{2} - \frac{1}{2}\right) = 0.$$

Case (d). In this case we select a move function for algorithm C such that $L_C(k) = L_B(k) \cup \{\sigma_k\}$, resulting in algorithm C not taking input k as a high selection, and giving an expected profit gain of

$$E[P_C - P_B] = \frac{n+1}{k!}\left(\frac{1}{2} - \frac{1}{k+1}\sigma_k(k)\right) \geq \frac{n+1}{k!}\left(\frac{1}{2} - \frac{1}{k+1}\cdot\frac{k+1}{2}\right) = 0.$$

In each case, we transformed algorithm B into a new algorithm C that performs at least as well (and hence must be optimal), and has a longer length of agreement with algorithm OP than B does. This directly contradicts our selection of B as the optimal algorithm with the longest length of agreement with OP, and this contradiction finishes the proof that algorithm OP is optimal. ∎

4. Conclusion. In this paper, we examined a natural on-line problem related to both financial games and the classic secretary problem. We select low and high values from a randomly ordered set of values presented in an on-line fashion, with the goal of maximizing the difference in final ranks of such low/high pairs. We considered two variations of this problem. The first allowed us to choose only a single low value followed by a single high value from a sequence of n values, while the second allowed selection of arbitrarily many low/high pairs. We presented provably optimal algorithms for both variants, gave tight analyses of the performance of these algorithms, and analyzed how well the on-line performance compares to the optimal off-line performance.

Our paper opens up many problems. Two particularly interesting directions are to consider more realistic input sources and to maximize quantities other than the difference in rank.

Appendix. Proof of expected final rank. In this appendix section, we prove that if an item has relative rank x_i among the first i inputs, then its expected rank r_i among all n inputs is given by $E[r_i|x_i] = \frac{n+1}{i+1}x_i$.

LEMMA A.1. *If a given item has rank x from among the first i inputs, and if the $i+1$st input is uniformly distributed over all possible rankings, then the expected rank of the given item among the first $i+1$ inputs is $\frac{i+2}{i+1}x$.*

Proof. If we let R be a random variable denoting the rank of our given item from among the first $i+1$ inputs, then we see that the value of R depends on the rank of the $i+1$st input. In particular, if the rank of the $i+1$st input is $\leq x$ (which happens with probability $\frac{x}{i+1}$), then the new rank of our given item will be $x+1$. On the other hand, if the rank of the $i+1$st input is $> x$ (which happens with probability $\frac{i+1-x}{i+1}$), then the rank of our given item is still x among the first $i+1$ inputs. Using this observation, we see that

$$E[R] = \frac{x}{i+1}(x+1) + \frac{i+1-x}{i+1}x = \frac{x+1+i+1-x}{i+1}x = \frac{i+2}{i+1}x,$$

which is what is claimed in the lemma. ∎

For a fixed position i, the above extension of rank to position $i+1$ is a constant times the rank of the item among the first i inputs. Because of this, we can simply extend this lemma to the case where x is not a fixed rank but is a random variable, and we know the expected rank among the first i items.

COROLLARY A.2. *If a given item has expected rank x from among the first i inputs, and if the $i+1$st input is uniformly distributed over all possible rankings, then the expected rank of the given item among the first $i+1$ inputs is $\frac{i+2}{i+1}x$.*

Simply multiplying together the change in expected rank from among i inputs, to among $i+1$ inputs, to among $i+2$ inputs, and so on up to n inputs, we get a telescoping product with cancellations between successive terms, resulting in the following corollary.

COROLLARY A.3. *If a given item has rank x from among the first i inputs, and if the remaining inputs are uniformly distributed over all possible rankings, then the expected rank of the given item among all n inputs is $\frac{n+1}{i+1}x$.*

REFERENCES

[1] M. AJTAI, N. MEGIDDO, AND O. WAARTS, *Improved algorithms and analysis for secretary problems and generalizations*, in Proc. 36th Symposium on Foundations of Computer Science, 1995, pp. 473–482.
[2] G. J. ALEXANDER AND W. F. SHARPE, *Fundamentals of Investments*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
[3] P. BILLINGSLEY, *Probability and Measure*, 2nd ed., John Wiley and Sons, New York, 1986.
[4] A. CHOU, J. COOPERSTOCK, R. EL-YANIV, M. KLUGERMAN, AND T. LEIGHTON, *The statistical adversary allows optimal money-making trading strategies*, in Proc. 6th Annual ACM-SIAM Symposium on Discrete Algorithms, 1995, pp. 467–476.
[5] Y. CHOW, H. ROBBINS, AND D. SIEGMUND, *Great Expectations: The Theory of Optimal Stopping*, Houghton Mifflin, Boston, 1971.
[6] Y. S. CHOW, S. MORIGUTI, H. ROBBINS, AND S. M. SAMUELS, *Optimal selection based on relative rank (the "secretary problem")*, Israel J. Math., 2 (1964), pp. 81–90.
[7] T. M. COVER, *An algorithm for maximizing expected log investment return*, IEEE Trans. Inform. Theory, IT-30 (1984), pp. 369–373.
[8] R. EL-YANIV, A. FIAT, R. KARP, AND G. TURPIN, *Competitive analysis of financial games*, in Proc. 33rd Symposium on Foundations of Computer Science, 1992, pp. 327–333.
[9] E. F. FAMA, *Foundations of Finance*, Basic Books, New York, 1976.
[10] P. R. FREEMAN, *The secretary problem and its extensions: A review*, Internat. Statist. Rev., 51 (1983), pp. 189–206.
[11] M. GARDNER, *Mathematical games*, Sci. Amer., 202 (1960), pp. 150–153.
[12] A. R. KARLIN, M. S. MANASSE, L. RUDOLPH, AND D. D. SLEATOR, *Competitive snoopy caching*, in Proc. 27th Symposium on Foundations of Computer Science, 1986, pp. 244–254.

[13] D. E. Knuth, *The Art of Computer Programming, Volume* 2: *Seminumerical Algorithms*, 2nd ed., Addison–Wesley, Reading, MA, 1981.
[14] A. Mucci, *Differential equations and optimal choice problems*, Ann. Statist., 1 (1973), pp. 104–113.
[15] W. T. Rasmussen and S. R. Pliska, *Choosing the maximum from a sequence with a discount function*, Appl. Math. Optim., 2 (1976), pp. 279–289.
[16] D. D. Sleator and R. E. Tarjan, *Amortized efficiency of list update and paging rules*, Comm. ACM, 28 (1985), pp. 202–208.
[17] M. H. Smith and J. J. Deely, *A secretary problem with finite memory*, J. Amer. Statist. Assoc., 70 (1975), pp. 357–361.
[18] M. Tamaki, *Recognizing both the maximum and the second maximum of a sequence*, J. Appl. Probab., 16 (1979), pp. 803–812.
[19] P. Whittle, *Optimization Over Time: Dynamic Programming and Stochastic Control: Volume* 1, John Wiley and Sons, Chichester, 1982.
[20] P. Wilmott, S. Howison, and J. Dewynne, *The Mathematics of Financial Derivatives*, Cambridge University Press, Cambridge, UK, 1995.

SIAM J. DISCRETE MATH.
Vol. 12, No. 1, pp. 91–110

SORTING PERMUTATIONS BY REVERSALS AND EULERIAN CYCLE DECOMPOSITIONS*

ALBERTO CAPRARA†

Abstract. We analyze the strong relationship among three combinatorial problems, namely, the problem of sorting a permutation by the minimum number of reversals (MIN-SBR), the problem of finding the maximum number of edge-disjoint alternating cycles in a breakpoint graph associated with a given permutation (MAX-ACD), and the problem of partitioning the edge set of an Eulerian graph into the maximum number of cycles (MAX-ECD). We first illustrate a nice characterization of breakpoint graphs, which leads to a linear-time algorithm for their recognition. This characterization is used to prove that MAX-ECD and MAX-ACD are equivalent, showing the latter to be NP-hard. We then describe a transformation from MAX-ACD to MIN-SBR, which is therefore shown to be NP-hard as well, answering an outstanding question which has been open for some years. Finally, we derive the worst-case performance of a well-known lower bound for MIN-SBR, obtained by solving MAX-ACD, discussing its implications on approximation algorithms for MIN-SBR.

Key words. sorting by reversals, breakpoint graph, Eulerian graph, cycle decomposition, complexity

AMS subject classifications. 68Q25, 68R10, 05C45

PII. S089548019731994X

1. Introduction.

Let $\pi = (\pi_1 \ \dots \ \pi_n)$ be a permutation of $\{1, \dots, n\}$, and denote by ι the identity permutation $(1 \ 2 \ \dots \ n-1 \ n)$. A *reversal* of the interval (i, j) is an inversion of the subsequence $\pi_i \ \dots \ \pi_j$ of π, represented by the permutation $\rho = (1 \ \dots \ i-1 \quad j \ \dots \ i \quad j+1 \ \dots \ n)$. Composition of π with ρ yields $\pi\rho = (\pi_1 \ \dots \ \pi_{i-1} \ \pi_j \ \dots \ \pi_i \ \pi_{j+1} \ \dots \ \pi_n)$, where elements $\pi_i, \dots, \pi_j$ have been reversed. The problem of *sorting a permutation by the minimum number of reversals* (MIN-SBR) is defined as follows.

MIN-SBR: Given a permutation π, find a shortest sequence of reversals $\rho_1, \dots, \rho_{d(\pi)}$ such that $\pi\rho_1 \dots \rho_{d(\pi)} = \iota$.
The optimal solution value $d(\pi)$ is called the *reversal distance* of π.

MIN-SBR was inspired by computational biology applications, in particular by genome rearrangements, and has been widely studied in the last years by Kececioglu and Sankoff [16, 15]; Bafna and Pevzner [1]; Hannenhalli and Pevzner [10, 11]; Caprara, Lancia, and Ng [6]; Berman and Hannenhalli [2]; Irving and Christie [13]; Tran [20]; Kaplan, Shamir, and Tarjan [14]; Christie [7]; and Caprara [5], among others.

Until recently, most evolutionary studies in molecular biology were based on sequence alignment, i.e., comparison of single genes to detect local mutations in the sequence of nucleotides. However, in the last few years, we have witnessed an increasing interest in analyzing entire genomes at once, thus shifting the attention from gene level to chromosome level (Sankoff et al. [19], Sankoff [18]). In fact, as it is often found

*Received by the editors April 21, 1997; accepted for publication (in revised form) August 12, 1998; published electronically January 29, 1999. This research was partially supported by MURST and CNR, Italy. A preliminary version of this paper appeared as *Sorting by reversals is difficult*, in Proc. First Annual International Conference on Computational Molecular Biology, RECOMB '97, Santa Fe, NM, 1997.

http://www.siam.org/journals/sidma/12-1/31994.html

†DEIS, Università di Bologna, Viale Risorgimento 2, I-40136 Bologna, Italy (acaprara@deis.unibo.it).

that the order of genes is preserved more easily than the DNA sequence (see [9, 21]), looking at genes instead of DNA sequences allows one to construct some otherwise extremely difficult evolutionary scenarios; this is particularly true for plant mitochondrial DNA, virology, and *Drosophila* genetics. Rearrangement of genomes can occur in many different ways, among which we can number inversions, transpositions, deletions, insertions, and duplications of fragments.

Let the order of the genes in two single-chromosome organisms be given by two permutations π and τ of $\{1, \ldots, n\}$. An inversion of the segment comprising the genes from the ith to the jth is represented by a reversal of the interval (i, j). A shortest sequence of reversals needed to transform π into τ is clearly equal to an optimal solution of MIN-SBR on $\tau^{-1}\pi$. Therefore, the solution of MIN-SBR yields a possible scenario to explain how an organism evolved from another, under the simplifying assumptions that inversions were the only rearrangement to occur, and that evolution required the minimum number of rearrangements. Even if these assumptions lead to some approximation, both are well motivated. Indeed, on the one hand inversions are by far the most frequent type of rearrangement, and on the other hand rearrangements are very rare events.

The breakpoint graph associated with a permutation has always played a key role in the study of MIN-SBR. In particular, the problem of finding a *maximum alternating-cycle decomposition* (MAX-ACD) of a breakpoint graph, defined in the next section, is very closely related to MIN-SBR. This strong relationship was first pointed out by Bafna and Pevzner [1] and Kececioglu and Sankoff [16] and was later considered by Hannenhalli and Pevzner [10, 11]. As will be clear from its definition, MAX-ACD is very similar to the following problem, called *maximum Eulerian cycle decomposition* (MAX-ECD).

MAX-ECD: Given an Eulerian graph $H = (W, E)$, find a maximum-cardinality cycle decomposition of H, i.e., partition E into the maximum number of cycles.

In this paper we further analyze the relationship between MIN-SBR, MAX-ACD, and MAX-ECD, deriving several results about the three problems.

One major open question about MIN-SBR is its complexity: although the problem was conjectured to be NP-hard back in 1993 by Kececioglu and Sankoff [16], nobody has been able to come up with a proof of this conjecture so far. In particular, the question appears as *Problem* 1 in the open problem list of Pevzner and Waterman [17] and is mentioned in the Crescenzi–Kann list of NP-hard problems; see entry MS9 in [8]. A stronger conjecture of Kececioglu and Sankoff [16] claimed that the special case of MIN-SBR, where, for a given π, one wants to check whether $d(\pi)$ is equal to one-half times the number of breakpoints of π (see below), was NP-complete. In fact Irving and Christie [13] and Tran [20] recently (and independently) disproved this latter conjecture, giving a polynomial time algorithm for solving this special case. Our results include a proof of the NP-hardness of MIN-SBR.

The paper is organized as follows. We illustrate the basic definitions and results from the literature in section 2. In section 3 we give a nice characterization of breakpoint graphs associated with permutations, yielding a linear-time algorithm for their recognition. We use this characterization in section 4 to prove that MAX-ACD is equivalent to MAX-ECD and is therefore NP-hard. In section 5 we describe a polynomial transformation from MAX-ACD to MIN-SBR, showing the latter is NP-hard. Finally, in section 6 we derive the absolute and asymptotic worst-case performance ratio of the lower bound on $d(\pi)$ obtained by solving MAX-ACD and discuss its implications on approximation algorithms for MIN-SBR.

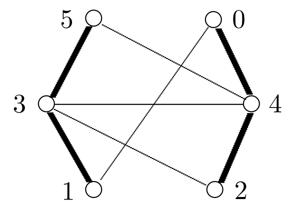


FIG. 1. *The breakpoint graph $G(\pi)$ associated with $\pi = (4\ 2\ 1\ 3)$. Gray edges are drawn as thin lines, black edges as thick lines.*

2. Basic definitions and previous results. In this section we give the basic definitions and previous results that we will use in what follows.

Consider a permutation $\pi = (\pi_1\ \dots\ \pi_n)$ of $\{1, \dots, n\}$. Following the description in [1], define the *breakpoint graph* $G(\pi) = (V, B \cup C)$ of π as follows. Add to π the elements $\pi_0 := 0$ and $\pi_{n+1} := n + 1$, redefining $\pi := (0\ \pi_1\ \dots\ \pi_n\ n + 1)$. Also, let the *inverse permutation* π^{-1} of π be defined by $\pi^{-1}_{\pi_i} := i$ for $i = 0, \dots, n + 1$. Let $V := \{0, \dots, n + 1\}$, where each node $v \in V$ represents an element of π. Graph $G(\pi)$ is *bicolored*, i.e., its edge set is partitioned into two subsets, each represented by a different color. B is the set of *black* edges, each of the form (π_i, π_{i+1}), for all $i \in \{0, \dots, n\}$ such that $|\pi_i - \pi_{i+1}| \neq 1$, i.e., elements which are in consecutive positions in π but not in the identity permutation ι. Such a pair π_i, π_{i+1} is called a *breakpoint* of π. Let $b(\pi) := |B|$ be the number of breakpoints of π. C is the set of *gray* edges, each of the form $(i, i+1)$, for all $i \in \{0, \dots, n\}$ such that $|\pi^{-1}_i - \pi^{-1}_{i+1}| \neq 1$, i.e., elements which are in consecutive positions in ι but not in π. Note that each node $i \in V$ has either degree 0, 2, or 4, and has the same number of incident gray and black edges. Therefore, $|B| = |C|(= b(\pi))$. Figure 1 depicts the breakpoint graph associated with the permutation (4 2 1 3).

An *alternating cycle* of $G(\pi)$ is a sequence of edges $b_1, c_1, b_2, c_2, \dots, b_m, c_m$, where $b_i \in B$, $c_i \in C$ for $i = 1, \dots, m$; b_i and c_j are incident to a common node for $i = j = 1, \dots, m$ and for $i = j + 1$, $j = 1, \dots, m$ (where $b_{m+1} := b_1$); and $b_i \neq b_j$, $c_i \neq c_j$ for $1 \leq i < j \leq m$. For example, edges $(0, 4), (4, 3), (3, 1), (1, 0)$ and $(4, 2), (2, 3), (3, 5), (5, 4)$ form alternating cycles in the graph of Figure 1. An *alternating path* is a subsequence of consecutive edges of some alternating cycle. It is sometimes convenient to assign each edge $(\pi_i, \pi_{i+1}) \in B$ an orientation from π_i to π_{i+1}, i.e., to orient the black edges of G from the endpoint which appears first in π to the endpoint which appears second. An alternating cycle of $G(\pi)$ is then called *unoriented with respect to* π if it is possible to walk along the whole cycle traversing each black edge in the direction of its orientation, *oriented with respect to* π otherwise. For example, in Figure 1 alternating cycle $(4, 2), (2, 3), (3, 5), (5, 4)$ is unoriented with respect to (4 2 1 3), whereas alternating cycle $(0, 4), (4, 3), (3, 1), (1, 0)$ is oriented with respect to (4 2 1 3).

An *alternating-cycle decomposition* of $G(\pi)$ is a collection of edge-disjoint alternating cycles, such that every edge of G is contained in exactly one cycle of the collection. It is easy to see that $G(\pi)$ always admits an alternating-cycle decomposition. In the graph of Figure 1, alternating cycles $(0, 4), (4, 3), (3, 1), (1, 0)$ and $(4, 2), (2, 3), (3, 5), (5, 4)$ form an alternating-cycle decomposition. For a given π let $c(\pi)$ be the maximum cardinality of an alternating-cycle decomposition of $G(\pi)$. Bafna and Pevzner [1] (see also Kececioglu and Sankoff [16]) proved the following property.

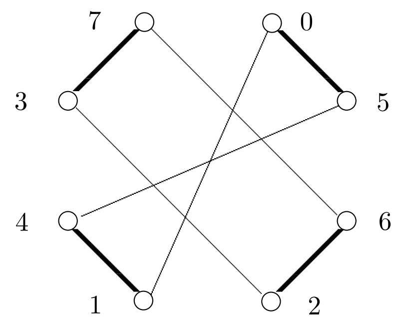


FIG. 2. *The breakpoint graph $G(\vec{\pi})$ associated with $\vec{\pi} = (3\ 1\ 2)$, where element 3 is even and elements 1 and 2 are odd.*

THEOREM 1 (See [1], [16]). *For every permutation π, $d(\pi) \geq b(\pi) - c(\pi)$.*

Therefore $b(\pi) - c(\pi)$ gives a valid lower bound on the optimal solution value to MIN-SBR. In practical cases this bound turns out to be very tight and is frequently equal to the optimum, as confirmed by the extensive experiments of Kececioglu and Sankoff [16] and Caprara, Lancia, and Ng [6]. This empirical observation was recently formalized by Caprara [5], who showed that the probability that $d(\pi) > b(\pi) - c(\pi)$ for a random permutation π of n elements is $O(1/n^5)$.

The *signed* version of MIN-SBR is a relevant variant of the problem, which was shown to be solvable in polynomial time by Hannenhalli and Pevzner [10]. This variant calls for sorting a permutation by the minimum number of reversals, with a *parity* assigned to each element of the permutation, specifying that, in a solution, the number of reversals involving the element must be either even, in which case the element is called *even*, or odd, in which case it is called *odd*. A permutation with a parity assigned to each element is called *signed* and is denoted by $\vec{\pi}$. Note that any sequence of reversals which sorts $\vec{\pi}$ corresponds to a feasible solution of MIN-SBR on π, the permutation obtained from $\vec{\pi}$ by neglecting the parity of the elements. Hence, letting $d(\vec{\pi})$ denote the optimal solution value of signed MIN-SBR on $\vec{\pi}$, one has that $d(\vec{\pi}) \geq d(\pi)$.

The breakpoints and the breakpoint graph of a signed permutation $\vec{\pi}$ with n elements correspond to those associated with the unsigned permutation ν with $2n$ elements defined by $\nu_{2i-1} := 2\vec{\pi}_i - 1$, $\nu_{2i} := 2\vec{\pi}_i$ if element $\vec{\pi}_i$ is even, $\nu_{2i-1} := 2\vec{\pi}_i$, $\nu_{2i} := 2\vec{\pi}_i - 1$ if element $\vec{\pi}_i$ is odd, for $i = 1, \ldots, n$. Accordingly, in the breakpoint graph $G(\vec{\pi})$ associated with $\vec{\pi}$, all nodes have degree 0 or 2; hence there is a unique alternating-cycle decomposition whose cardinality is denoted by $c(\vec{\pi})$. Bafna and Pevzner showed that Theorem 1 also applies to signed permutations, i.e., $d(\vec{\pi}) \geq b(\vec{\pi}) - c(\vec{\pi})$, where $b(\vec{\pi})$ is the number of breakpoints of $\vec{\pi}$, equal to $b(\nu)$. Figure 2 depicts the breakpoint graph associated with $\vec{\pi} = (3\ 1\ 2)$ where element 3 is even and elements 1 and 2 are odd, hence $\nu = (5\ 6\ 2\ 1\ 4\ 3)$.

The following definitions follow those of Hannenhalli and Pevzner [10]. The orientation of the black edges of $G(\vec{\pi})$ with respect to $\vec{\pi}$ is defined as their orientation with respect to ν. Two gray edges (ν_i, ν_h) and (ν_j, ν_k) of $G(\vec{\pi})$ are called *interleaving* if their endpoints are such that either $i < j < h < k$ or $i > j > h > k$. In the example of Figure 2, gray edges $(0, 1)$ and $(6, 7)$ are interleaving, while edges $(6, 7)$ and $(2, 3)$ are not. Two alternating cycles A_1 and A_2 of $G(\vec{\pi})$ are called *interleaving* if there are two interleaving edges $e_1 \in A_1$ and $e_2 \in A_2$. The *interleaving graph* $H(\vec{\pi})$ associated with $\vec{\pi}$ has one node A associated with each alternating cycle A of $G(\vec{\pi})$ and one edge (A_1, A_2) associated with each pair of interleaving cycles A_1 and A_2. The interleaving graph associated with the signed permutation of Figure 2 has two nodes connected by an edge as edges $(0, 1)$ and $(6, 7)$ are interleaving.

Hannenhalli and Pevzner [10] define the parameters $h(\vec{\pi})$ and $f(\vec{\pi})$, both related to the structure of $H(\vec{\pi})$. For the scope of this paper, we avoid giving the definition of $h(\vec{\pi})$ and $f(\vec{\pi})$ but simply stress that $h(\vec{\pi}) + f(\vec{\pi})$ is at most equal to the number of connected components of $H(\vec{\pi})$ in which all nodes correspond to unoriented alternating cycles. The main result proved by Hannenhalli and Pevzner in [10] is in the following.

THEOREM 2 (See [10]). *For every signed permutation* $\vec{\pi}$, $d(\vec{\pi}) = b(\vec{\pi}) - c(\vec{\pi}) + h(\vec{\pi}) + f(\vec{\pi})$.

As a corollary, one gets that $d(\vec{\pi}) = b(\vec{\pi}) - c(\vec{\pi})$ if every alternating cycle of $G(\vec{\pi})$ is oriented. We note that this condition is only sufficient. The above theorem leads to a polynomial time algorithm for signed MIN-SBR. At present, the most efficient algorithm is due to Kaplan, Shamir, and Tarjan [14].

THEOREM 3 (See [14]). *The signed version of MIN-SBR can be solved in* $O(n^2)$ *time.*

Signed permutations can be used to establish an elegant connection between alternating-cycle decompositions and solutions of MIN-SBR, as illustrated next.

Given a (unsigned) permutation π and an alternating-cycle decomposition of $G(\pi)$ into, say, p cycles, it is easy to assign a parity to the elements of π so that the resulting signed permutation $\vec{\pi}$ satisfies $b(\vec{\pi}) = b(\pi)$ and $c(\vec{\pi}) = p$. The main idea is to decide whether each element of $\vec{\pi}$ must be even or odd so as to ensure that the unique alternating-cycle decomposition of $G(\vec{\pi})$ coincides with the given alternating-cycle decomposition of $G(\pi)$; see [11, 6] for details. For our purposes, it is important to observe that every alternating cycle of $G(\vec{\pi})$ is oriented with respect to $\vec{\pi}$ if and only if its counterpart in the decomposition of $G(\pi)$ is oriented with respect to π. Therefore, assuming that $p = c(\pi)$, i.e., the given alternating-cycle decomposition is optimal, and that every cycle in the decomposition is oriented with respect to π, one has

$$d(\pi) \le d(\vec{\pi}) = b(\vec{\pi}) - c(\vec{\pi}) = b(\pi) - c(\pi) \le d(\pi),$$

hence implying the following theorem.

THEOREM 4 (See [10, 11]). *For a permutation* π *and a maximum decomposition of* $G(\pi)$ *into* $c(\pi)$ *alternating cycles, if every cycle in the decomposition is oriented with respect to* π*, then* $d(\pi) = b(\pi) - c(\pi)$.

The above discussion motivates the study of the following MAX-ACD problem.

MAX-ACD: Given the breakpoint graph $G(\pi)$ of a given permutation π, find a maximum-cardinality alternating-cycle decomposition of $G(\pi)$.

As one can observe, MAX-ACD is somehow related to MAX-ECD, defined in the previous section. In the early 1980s Holyer [12] proved that checking whether the edge set of a given graph H can be partitioned into cliques of size k is NP-complete for every $k \ge 3$. In particular, for $k = 3$ one wants to check whether the edge set of H can be partitioned into triangles. In this case H can be assumed to be Eulerian without loss of generality, the answer clearly being no otherwise. So the problem of determining whether the edge set of an Eulerian graph can be partitioned into triangles is NP-complete. This immediately implies the following theorem.

THEOREM 5 (See [12]). *MAX-ECD is NP-hard.*

3. A nice characterization of breakpoint graphs. In this section we characterize the bicolored graphs which are breakpoint graphs of some permutation. In particular, we derive a linear-time algorithm which checks whether a given bicolored graph is a breakpoint graph and, if this is the case, yields an associated permutation. These results will be used in the next sections.

To simplify the notation, given a (possibly bicolored) graph $G = (V, E)$ and any set $F \subseteq \{(i, j) : i, j \in V, i \neq j\}$ (possibly $F \not\subseteq E$), we let $G(F)$ denote the subgraph of G induced by F, defined by node set $V \setminus I$, where I is the set of nodes not contained in any pair in F, and edge set F. Furthermore, we write $G_1 = G_2$ to indicate that two graphs G_1 and G_2 are isomorphic.

Definition 1. *A bicolored graph* $G = (V, B \cup C)$ *is called* balanced bicolored *if*

(i) *each connected component of the subgraphs of* G *induced by edge set* B, $G(B)$, *and by edge set* C, $G(C)$, *is a simple path;*
(ii) *each node* $i \in V$ *has the same degree* $(0, 1,$ *or* $2)$ *in* $G(B)$ *and* $G(C)$;
(iii) *the edge sets of* $G(B)$ *and* $G(C)$ *are disjoint; i.e.,* $B \cap C = \emptyset$.

Let $G = (V, B \cup C)$ be a balanced bicolored graph. For convenience, in this section we do not address explicitly the trivial case $B \cup C = \emptyset$. Consider the set U of the nodes of degree 2 in G, i.e., the set of nodes of degree 1 in $G(B)$ and $G(C)$. Note that $|U|$ is even and ≥ 2. A *perfect matching* of the nodes in U is a set $M \subset \{(i, j) : i, j \in U, i \neq j\}$ such that every $i \in U$ is contained in exactly one pair in M. Such a perfect matching is called *Hamiltonian* if the graphs $G(B \cup M)$ and $G(C \cup M)$ are Hamiltonian circuits; note that M can be Hamiltonian only if $M \cap (B \cup C) = \emptyset$ and that both $G(B \cup M)$ and $G(C \cup M)$ visit only the nodes of degree 1 or 2 in G. The importance of Hamiltonian matchings of U is motivated by the following property.

Lemma 1. *Let* G *be a balanced bicolored graph and* U *be the set of degree-*2 *nodes of* G. *Every Hamiltonian matching of* U *defines at least one permutation* π *such that* $G(\pi) = G$. *Conversely, every* π *such that* $G(\pi) = G$ *defines a Hamiltonian matching of* U.

Proof. Given a Hamiltonian matching M of U, a permutation π of $|V|$ elements (including the dummy elements $\pi_0 := 0$ and $\pi_{|V|-1} := |V| - 1$) such that $G(\pi) = G$ can be constructed as follows. First of all, if G has no node of degree 0, let $P := M$. Otherwise, let $i_1, \ldots, i_k$ denote the nodes of degree 0 in G, arbitrarily choose $(i, j) \in M$, and let $P := (M \setminus \{(i, j)\}) \cup \{(i, i_1), (i_1, i_2), \ldots, (i_{k-1}, i_k), (i_k, j)\}$; i.e., P is obtained from M by replacing edge (i, j) with the edges in a path from i to j visiting all the nodes of degree 0 in G. Both $G(B \cup P)$ and $G(C \cup P)$ are Hamiltonian circuits visiting all the nodes of G. Then, choose any node $i \in U$ and number it 0. Consider the Hamiltonian circuit $G(C \cup P)$, and walk along it starting from i, first traversing the gray edge incident to i. Number the nodes $1, 2, \ldots, |V| - 1$ according to the order in which they are visited by the walk. Now consider the Hamiltonian circuit $G(B \cup P)$, and walk along it starting from node i, letting $\pi_0 := 0$, and first traversing the black edge incident to i. Let elements $\pi_1, \pi_2, \ldots, \pi_{|V|-1}$ correspond to the numbers assigned to the nodes which are in turn visited by the walk. It is easy to verify that $G(\pi) = G$.

Conversely, for any given permutation π of $\{1, \ldots n\}$ such that $G(\pi) = G$, let $P := (\{(i, i+1) : 0 \leq i \leq n\} \setminus C) \cup \{(0, n+1)\} = (\{(\pi_i, \pi_{i+1}) : 0 \leq i \leq n\} \setminus B) \cup \{(0, n+1)\}$. It is easy to verify that $G(P)$ is a set of paths whose endpoints are the nodes of degree 2 in G and whose intermediate nodes are the nodes of degree 0 in G. Moreover, the set M containing all pairs (i, j) such that there is a maximal path from i to j in $G(P)$

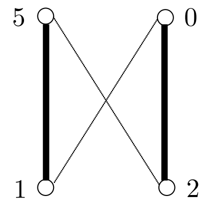


FIG. 3. *The auxiliary graph A associated with the breakpoint graph of Figure* 1. *Edges in E are drawn as thin lines, edges in D as thick lines.*

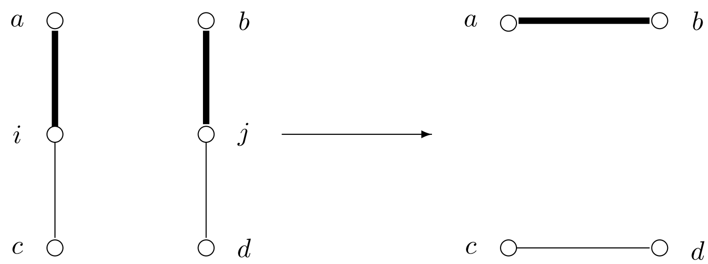


FIG. 4. *The removal of nodes i and j before each recursive call of* HAMILTONIAN_MATCHING.

is a Hamiltonian matching of G. ∎

For example, the Hamiltonian matching of $G(\pi)$ in Figure 1 defined by π is $M = \{(1, 2), (0, 5)\}$. Matching M is also associated with the permutation (2 4 3 1), whose breakpoint graph is isomorphic to $G(\pi)$.

Note that given a breakpoint graph G and an associated Hamiltonian matching M, the Hamiltonian circuit $G(B \cup M)$ uniquely determines which alternating cycles of G are unoriented or oriented with respect to every permutation associated with G and M.

We show that every balanced bicolored graph has a Hamiltonian matching by giving a procedure to find such a matching. Given a balanced bicolored graph G, construct the following *auxiliary graph* $A = (U, D \cup E)$, where nodes $i, j \in U$ are connected by a black edge $d \in D$ if there is a maximal path in $G(B)$ from i to j, and, symmetrically, nodes $i, j \in U$ are connected by a gray edge $e \in E$ if there is a maximal path in $G(C)$ from i to j. It is easy to verify that each node of A has exactly 2 incident edges, one in D and one in E. (Note that A is not necessarily a balanced bicolored graph, since a black edge in D and a gray edge in E might have the same endpoints.) Figure 3 depicts the auxiliary graph associated with the breakpoint graph of Figure 1. Consider procedure HAMILTONIAN_MATCHING of Figure 5 applied to A.

LEMMA 2. *The set M constructed by procedure* HAMILTONIAN_MATCHING *is a Hamiltonian matching.*

Proof. If $|U| > 2$, i.e., $|U| \geq 4$, there always exists a node pair i, j such that $(i, j) \notin D \cup E$, since each node of A has degree 2. The fact that M is a perfect matching of the nodes of degree 2 in G such that $M \cap (B \cup C) = \emptyset$ is immediate. To see that M is Hamiltonian, it is sufficient to show that $G(B \cup M)$ and $G(C \cup M)$ contain no subcircuit. Indeed, in each recursive step, there is a one-to-one correspondence between maximal paths in $G(B \cup M)$ (resp., in $G(C \cup M)$) and edges in D (resp., in E), and therefore the only way to introduce a subcircuit would be by matching two

```
procedure HAMILTONIAN_MATCHING(A, M);
input: the auxiliary graph A = (U, D ∪ E) associated with
    a balanced bicolored graph G = (V, B ∪ C);
output: a Hamiltonian matching M of the nodes in U;
begin
    if |U| = 2 then
        let i and j be the two nodes in U;
        let M := {i, j}
    else
    begin
        choose any node pair i, j ∈ U such that (i, j) ∉ D ∪ E;
        comment: remove i and j (see Figure 4);
        let a and b respectively be the nodes connected to i and j
        by a black edge in D, and c and d respectively be the nodes
        connected to i and j by a gray edge in E;
        let U := U \ {i, j}, D := (D \ {(i, a), (j, b)}) ∪ {(a, b)},
        E := (E \ {(i, c), (j, d)}) ∪ {(c, d)}, A := (U, D ∪ E);
        call HAMILTONIAN_MATCHING(A, M);
        let M := M ∪ {(i, j)}
    end
end.
```

FIG. 5. *Procedure* HAMILTONIAN_MATCHING.

nodes connected by an edge in D (resp., in E), which is avoided. ∎

For graph $G(\pi)$ in Figure 1, $M = \{(1,2),(0,5)\}$ is the only possible Hamiltonian matching; see Figure 3.

From the above lemmas we get the following theorem.

THEOREM 6. *Every balanced bicolored graph is isomorphic to the breakpoint graph of some permutation, and vice versa.*

Proof. Every breakpoint graph satisfies, by definition, (i)–(iii) in Definition 1. Conversely, given a balanced bicolored graph G, by Lemma 2 it is possible to find a Hamiltonian matching of G and hence, by Lemma 1, a permutation π such that $G(\pi) = G$. ∎

The above discussion leads to a linear-time recognition algorithm for breakpoint graphs.

COROLLARY 1. *Given a bicolored graph $G = (V, B \cup C)$, it is possible to check in $O(|V|)$ time whether G is a breakpoint graph, obtaining an associated permutation if the answer is positive.*

Proof. Suppose the given bicolored graph $G = (V, B \cup C)$ is stored using one vector of length $|V|$ containing the number of black and gray edges incident to each node, and two vectors of length $2|V|$, one for each edge color, where entries $2i-1$ and $2i$ contain the nodes connected to node i by gray and black edges, respectively. (If a node turns out to have a different number of incident gray and black edges, or more

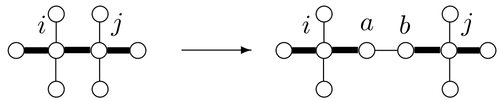


FIG. 6. *The subdivision of edge* (i, j).

than two incident black/gray edges, G is clearly not a breakpoint graph.) It is then straightforward to check if G is balanced bicolored in $O(|V|)$ time. Furthermore, the auxiliary graph A can easily be constructed and stored in the same way as G in $O(|V|)$ time. With this data structure for A, it is obvious how to perform each recursive step in HAMILTONIAN_MATCHING in constant time. In particular, for any node i of the auxiliary graph, by checking at most three other nodes it is possible to find a node j not connected to i. Finally, the construction of a permutation associated with G can be done in $O(|V|)$ time if one adds to the data structure for G a vector of length $|U|$ storing, for each node i of degree 2 in G, the node connected to i by an edge in the Hamiltonian matching. ∎

Given a bicolored graph $G = (V, B \cup C)$, the *subdivision* of an edge $e = (i, j) \in B \cup C$ is obtained by adding two new nodes, say a, b, to V, and replacing e by the 3 edges (i, a), (a, b), and (b, j), where (i, a) and (b, j) have the same color as e, while (a, b) has a different color from e; see Figure 6.

REMARK 1. *There is a one-to-one correspondence between alternating cycles (and alternating-cycle decompositions) of a bicolored graph G and any bicolored graph G' obtained from G by subdividing edges.*

Edge subdivision and analogous operations in breakpoint graphs have been used in the literature for different purposes; see, for example, [10] and [6]. In this paper, edge subdivision plays an important role in the analysis of the complexity of MIN-SBR in section 5.

4. The equivalence of MAX-ECD and MAX-ACD. In this section we show that MAX-ECD and MAX-ACD can be rather easily transformed into each other. The main implications of our constructions are that MAX-ACD is NP-hard, that any approximation algorithm for MAX-ACD with a given worst-case performance can be used for solving MAX-ECD with the same worst-case performance, and that MAX-ECD is NP-hard even if restricted to Eulerian graphs with degree at most 4.

The transformation from MAX-ACD to MAX-ECD works as follows. Let $G = (V, B \cup C)$ be any balanced bicolored graph. Define the Eulerian graph H_G from G by replacing every degree-4 node in V by the graph $X(i)$ in Figure 7. Formally, each degree-4 node $i \in V$ having incident black edges (i, a) and (i, c) and incident gray edges (i, b) and (i, d) is replaced by nodes i_0, i_1, i_2, i_3, i_4, by (uncolored) edges $(i_0, i_1), (i_0, i_2), (i_0, i_3), (i_0, i_4), (i_1, i_2), (i_2, i_3), (i_3, i_4), (i_4, i_1)$, by black edges (i_1, a), (i_3, c), and by gray edges $(i_2, b), (i_4, d)$. After all the original degree-4 nodes of G have been replaced, graph H_G is obtained by simply forgetting about the color of the edges. Note that every node of H_G has degree at most 4. Letting s be the number of degree-4 nodes in G, it would be easy to show that there is a one-to-one correspondence between alternating-cycle decompositions of G into p cycles and cycle decompositions of H_G into $p + 2s$ cycles, $2s$ of which are triangles internal to the $X(i)$'s, proving the following.

THEOREM 7. *Given a balanced bicolored graph G, MAX-ACD on G can be solved*

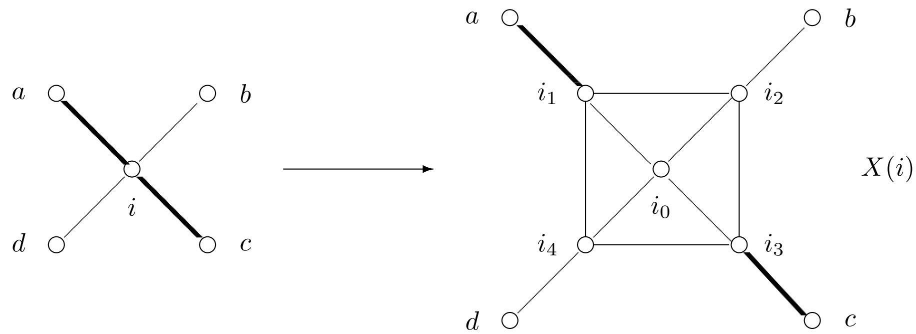


FIG. 7. *The replacement of each degree-4 node i by graph $X(i)$ in the transformation from MAX-ACD to MAX-ECD.*

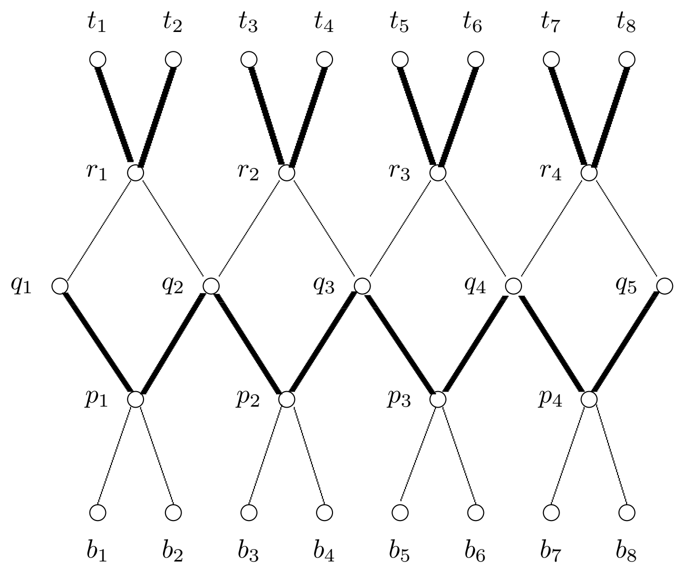


FIG. 8. *Graph $Y(8)$.*

by solving MAX-ECD on graph H_G, whose size is linear in the size of G and where every node has degree at most 4.

For a detailed proof, see the original technical report [3]. Unfortunately, the transformation presented does not preserve approximability, since the optimal solutions of MAX-ACD on G and MAX-ECD on H_G differ by twice the number of degree-4 nodes in G.

In order to show a polynomial transformation of MAX-ECD into MAX-ACD, we need some preliminary definitions and lemmas.

Let d be an even integer and $s := d/2$. The bicolored graph $Y(d)$ is defined by node set $\{b_1, \ldots, b_d\} \cup \{p_1, \ldots, p_s\} \cup \{q_1, \ldots, q_{s+1}\} \cup \{r_1, \ldots, r_s\} \cup \{t_1, \ldots, t_d\}$, by black edge set $\{(p_i, q_i), (p_i, q_{i+1}), i = 1, \ldots, s\} \cup \{(r_i, t_{2i-1}), (r_i, t_{2i}), i = 1, \ldots, s\}$, and by gray edge set $\{(b_{2i-1}, p_i), (b_{2i}, p_i), i = 1, \ldots, s\} \cup \{(q_i, r_i), (q_{i+1}, r_i), i = 1, \ldots, s\}$. Graph $Y(8)$ is depicted in Figure 8.

Let $Z(d, m)$ be the bicolored graph obtained by merging m copies of

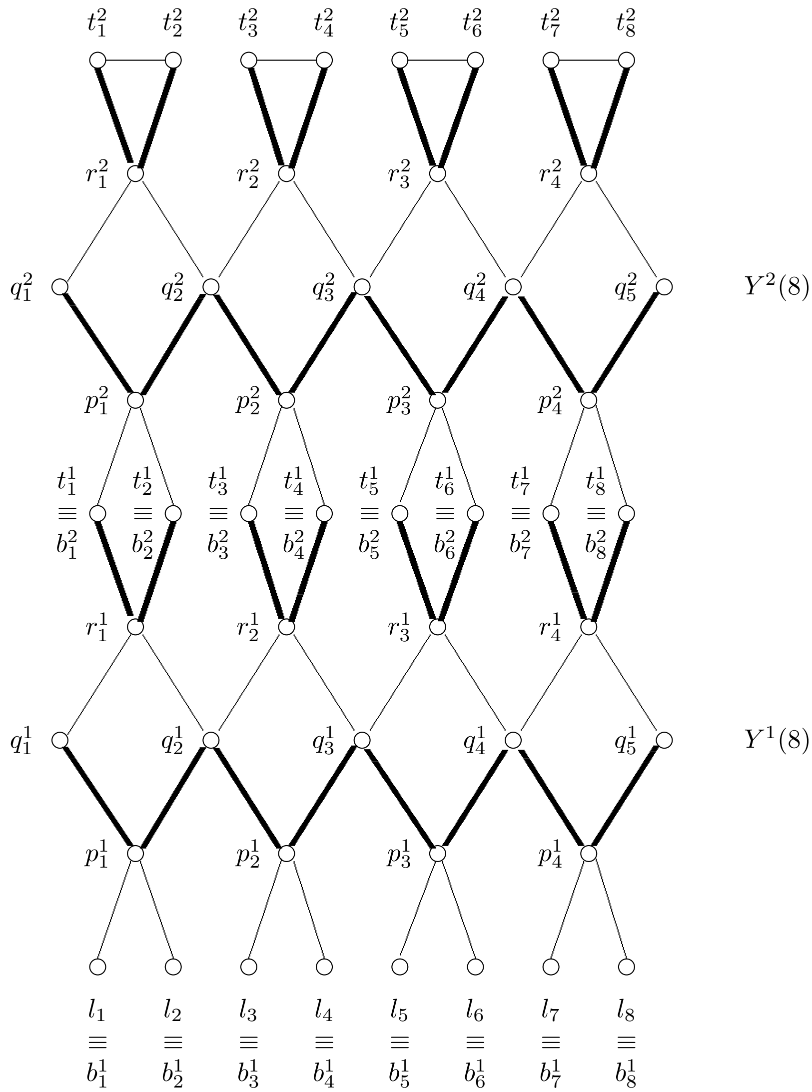


FIG. 9. *Graph* $Z(8,2)$.

$Y(d)$, say $Y^1(d),\dots,Y^m(d)$, as follows. For $j = 1,\dots,m-1$ the t-nodes of $Y^j(d)$ and the b-nodes of $Y^{j+1}(d)$ are identified; more precisely node t_i in $Y^j(d)$ and node b_i in $Y^{j+1}(d)$ $(i = 1,\dots,d)$ correspond to the same node in $Z(d,m)$, connected to nodes $p_{\lceil i/2\rceil}$ in $Y^j(d)$ and $r_{\lceil i/2\rceil}$ in $Y^{j+1}(d)$. Moreover, the t-nodes in $Y^m(d)$ are joined in $Z(d,m)$ by gray edges $\{(t_{2i-1},t_{2i}), i = 1,\dots,s\}$. Graph $Z(8,2)$ is shown in Figure 9.

Let $l_1,\dots,l_d$ represent the nodes in $Z(d,m)$ corresponding to nodes $b_1,\dots,b_d$ in $Y^1(d)$. Moreover, let b_i^j, p_i^j, q_i^j, r_i^j, and t_i^j denote, respectively, the nodes in $Z(d,m)$ corresponding to nodes b_i, p_i, q_i, r_i, and t_i in $Y^j(d)$. In particular, for $i = 1,\dots,d$ and $j = 1,\dots,m-1$, nodes t_i^j and b_i^{j+1} coincide. Also, node sets $l_1,\dots,l_d$ and $b_1^1,\dots,b_d^1$ coincide. For notational convenience we represent alternating paths in $Z(d,m)$ by the sequence of nodes they visit. The *trivial alternating path* which connects nodes b_{2i-1}^m

and b_{2i}^m in $Z(d,m)$ is given by

$$b_{2i-1}^m, p_i^m, q_i^m, r_i^m, t_{2i-1}^m, t_{2i}^m, r_i^m, q_{i+1}^m, p_i^m, b_{2i}^m;$$

see Figure 9. The next lemma follows directly from the definition of $Z(d,m)$.

LEMMA 3. *$Z(d,m)$ does not contain any alternating cycle, each node in $Z(d,m)$ has the same number of incident gray and black edges, with the exception of $l_1,\dots,l_d$, and any alternating path connecting two l-nodes contains exactly $8m+1$ edges.*

We are now ready to prove the key property of $Z(d,m)$.

LEMMA 4. *Let τ be any partition of the node set $\{l_1,\dots,l_d\}$ in $Z(d,m)$ into pairs, say $\tau=\{(l_{\tau_1},l_{\tau_2}),\dots,(l_{\tau_{d-1}},l_{\tau_d})\}$. If $m\ge 1+s(s-1)/2$, the edge set of $Z(d,m)$ can be decomposed into s alternating paths, each connecting a different pair in τ.*

Proof. Note that every partition τ can be obtained from the partition $\{(l_1,l_2),\dots,(l_{d-1},l_d)\}$ by performing at most $s(s-1)/2$ exchanges between adjacent pairs. Indeed, with at most $s-1$ exchanges one can get the first pair in τ, with at most $s-2$ the second, and so on. Our proof is by induction on the number of such exchanges. We show that, for $i=1,\dots,s$, it is possible to connect nodes $l_{\tau_{2i-1}}$ and $l_{\tau_{2i}}$ by an alternating path of the form $l_{\tau_{2i-1}},\dots,t_{2i-1}^m,t_{2i}^m,\dots,l_{\tau_{2i}}$.

Clearly, if $\tau=\{(l_1,l_2),\dots,(l_{d-1},l_d)\}$, the edge set of $Z(d,1)$ can be decomposed as required by using, for $i=1,\dots,s$, the trivial alternating path which connects l_{2i-1} to l_{2i}.

Suppose now that τ is obtained from τ' by exchanging two elements in consecutive pairs, and the edge set of $Z(d,j)$ can be decomposed into s alternating paths, the ith of the form $l_{\tau'_{2i-1}},\dots,t_{2i-1}^j,t_{2i}^j,\dots l_{\tau'_{2i}}$. Then, the edge set of $Z(d,j+1)$ can be decomposed into s alternating paths, the ith of the form $l_{\tau_{2i-1}},\dots,t_{2i-1}^{j+1},t_{2i}^{j+1},\dots l_{\tau_{2i}}$. These alternating paths are obtained by taking the alternating paths in $Z(d,j)$ corresponding to τ' and replacing their gray edges (t_{2i-1}^j,t_{2i}^j), $i=1,\dots,s$, by alternating paths in $Z(d,j+1)$ using edges of $Y^{j+1}(d)$ and edges $(t_{2i-1}^{j+1},t_{2i}^{j+1})$, $i=1,\dots,s$.

For completeness, we consider the two possible cases. In the first case, pairs $(l_{\tau'_{2h-1}},l_{\tau'_{2h}})$ and $(l_{\tau'_{2h+1}},l_{\tau'_{2h+2}})$ are replaced in τ by pairs $(l_{\tau'_{2h-1}},l_{\tau'_{2h+1}})$ and $(l_{\tau'_{2h}}, l_{\tau'_{2h+2}})$. The alternating paths corresponding to τ are obtained from those corresponding to τ' by replacing each edge (t_{2i-1}^j,t_{2i}^j) in $Z(d,j)$, $i=1,\dots,h-1$ and $i=h+2,\dots,s$, by the trivial alternating path connecting nodes b_{2i-1}^{j+1} and b_{2i}^{j+1} in $Z(d,j+1)$ and by replacing edges (t_{2h-1}^j,t_{2h}^j) and (t_{2h-1}^j,t_{2h}^j) in $Z(d,m)$ by the alternating paths

$$b_{2h-1}^{j+1},p_h^{j+1},q_h^{j+1},r_h^{j+1},t_{2h-1}^{j+1},t_{2h}^{j+1},r_h^{j+1},q_{h+1}^{j+1},p_{h+1}^{j+1},b_{2h+1}^{j+1}$$

and

$$b_{2h}^{j+1},p_h^{j+1},q_{h+1}^{j+1},r_{h+1}^{j+1},t_{2h+1}^{j+1},t_{2h+2}^{j+1},r_{h+1}^{j+1},q_{h+2}^{j+1},p_{h+1}^{j+1},b_{2h+2}^{j+1}.$$

In the second case, pairs $(l_{\tau'_{2h-1}},l_{\tau'_{2h}})$ and $(l_{\tau'_{2h+1}},l_{\tau'_{2h+2}})$ are replaced in τ by pairs $(l_{\tau'_{2h-1}},l_{\tau'_{2h+2}})$ and $(l_{\tau'_{2h}},l_{\tau'_{2h+1}})$, and the construction is analogous to the previous one.

The proof is completed by observing that if the edge set of $Z(d,j)$ can be decomposed as required, the edge set of $Z(d,j+1)$ can as well. Indeed, it is sufficient to replace edges (t_{2i-1}^j,t_{2i}^j), $i=1,\dots,s$, in $Z(d,j)$ by the trivial alternating paths connecting nodes b_{2i-1}^{j+1} and b_{2i}^{j+1} in $Z(d,j+1)$. ☐

Given an even integer d, let $m(d):=1+d/2(d/2-1)/2$, observing that $m(d)$ is integer. The transformation from MAX-ECD to MAX-ACD, illustrated in Figure 10,

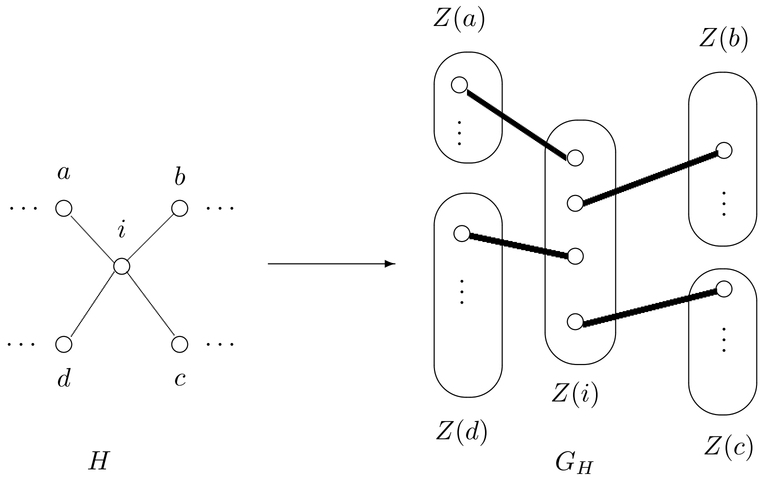


FIG. 10. *Outline of the transformation from MAX-ECD to MAX-ACD. The circles inside each $Z(v)$ represent the associated degree-1 nodes.*

is then the following. Given an Eulerian graph $H = (W, E)$, the bicolored graph G_H is obtained from H by replacing each $i \in W$, of degree d_i, by the graph $Z(i) := Z(d_i, m(d_i))$ and then, for each $(i, j) \in F$, by connecting two degree-1 nodes of $Z(i)$ and $Z(j)$ by a black edge. This connection is easily made so that all degree-1 nodes of $Z(i)$, $i \in W$, have exactly one incident black edge. Note that the size of $Z(i)$ is proportional to d_i^3. It is then clear by Lemma 4 that given a cycle decomposition of H into p cycles we can find an alternating-cycle decomposition of G_H into p cycles, and vice versa. Finally, G_H is clearly balanced bicolored.

To summarize, we have proved the following theorem.

THEOREM 8. *Given an Eulerian graph H, MAX-ECD on H can be solved by solving MAX-ACD on the balanced bicolored graph G_H, whose size is linear in the size of H and in d_H^3, where d_H is the maximum degree of a node of H.*

From Theorems 5, 6, and 8 we immediately get the following result.

COROLLARY 2. *MAX-ACD is NP-hard.*

The following remark stresses a property of the above transformation from MAX-ECD to MAX-ACD.

REMARK 2. *The above transformation from MAX-ECD to MAX-ACD preserves approximability, namely, to each solution of MAX-ECD on H there corresponds a solution of MAX-ACD on G_H of the same value, and vice versa.*

Unfortunately, we are not aware of any approximability result for MAX-ACD (and MAX-ECD); see also the next section.

We conclude this section with another NP-hardness result, which is a consequence of Theorem 7 and Corollary 2.

COROLLARY 3. *MAX-ECD restricted to Eulerian graphs with maximum degree 4 is NP-hard.*

5. The complexity of MIN-SBR. In this section we give a transformation from MAX-ACD to MIN-SBR, showing the latter is NP-hard. The transformation makes extensive use of the breakpoint graph characterization illustrated in section 3.

A possible way of reading Theorem 4 is the following. Given a breakpoint graph $G(\pi)$ of some permutation π, it would be possible to compute $c(\pi)$ by solving MIN-

procedure DOUBLE_SUBDIVISION(G, M, G^*, M^*);
input: a breakpoint graph $G = (V, B \cup C)$ and an associated
  Hamiltonian matching M of the nodes of degree 2 in G;
output: a breakpoint graph $G^* = (V^*, B^* \cup C^*)$ and an associated
  Hamiltonian matching M^* of the nodes of degree 2 in G^*
  such that there is a one-to-one correspondence between
  alternating cycles (and alternating-cycle decompositions) of G and G^*
  and all alternating cycles of G^* are oriented with respect to every
  permutation π^* associated with G^* and M^*;
begin
  initialize $V^* := V$, $B^* := B$, $C^* := C$, $M^* := M$;
  for each $r \in B$ **do**
  begin
    let i and j be the endpoints of r and $e = (a, b)$ be any edge of M^*,
    where a and b are such that the Hamiltonian circuit $G^*(B^* \cup M^*)$ is
    the union of edge e, a path from a to i, edge r, and a path from j to b
    (possibly $i = a$ or $j = b$, but not both);
    comment: add new nodes k_1, k_2, k_3, k_4 to V^*, and replace edge (i, j)
      by the alternating path $(i, k_1), (k_1, k_2), (k_2, k_3), (k_3, k_4), (k_4, j)$;
    let $V^* := V^* \cup \{k_1, k_2, k_3, k_4\}$,
    $B^* := (B^* \setminus \{(i, j)\}) \cup \{(i, k_1), (k_2, k_3), (k_4, j)\}$,
    $C^* := C^* \cup \{(k_1, k_2), (k_3, k_4)\}$;
    comment: update M^* so as to define a Hamiltonian matching on the
      modified G^*, such that all alternating cycles containing the new
      black edges are oriented with respect to every permutation
      associated with G^* and M^*;
    let $M^* := (M^* \setminus \{(a, b)\}) \cup \{(k_1, k_4), (k_2, a), (k_3, b)\}$
  end
end.

FIG. 11. *Procedure* DOUBLE_SUBDIVISION.

SBR on π if there existed an optimal alternating-cycle decomposition of $G(\pi)$ made up of oriented (with respect to π) cycles only. Unfortunately, this is not always the case; we show how to overcome this difficulty in what follows.

Consider a breakpoint graph $G(\pi) = (V, B \cup C)$ associated with some permutation π, and let M be the Hamiltonian matching of $G(\pi)$ determined by π (see section 3). Construct the breakpoint graph G^* and its Hamiltonian matching M^* by applying to $G(\pi)$ and M the procedure DOUBLE_SUBDIVISION in Figure 11. This procedure replaces every black edge of G by an alternating path of 5 edges, "twice" subdividing the original edge. Figure 12 shows the effect of the replacement of the black edge (i, j) on graphs $G^*(B^* \cup M^*)$ and $G^*(C^* \cup M^*)$.

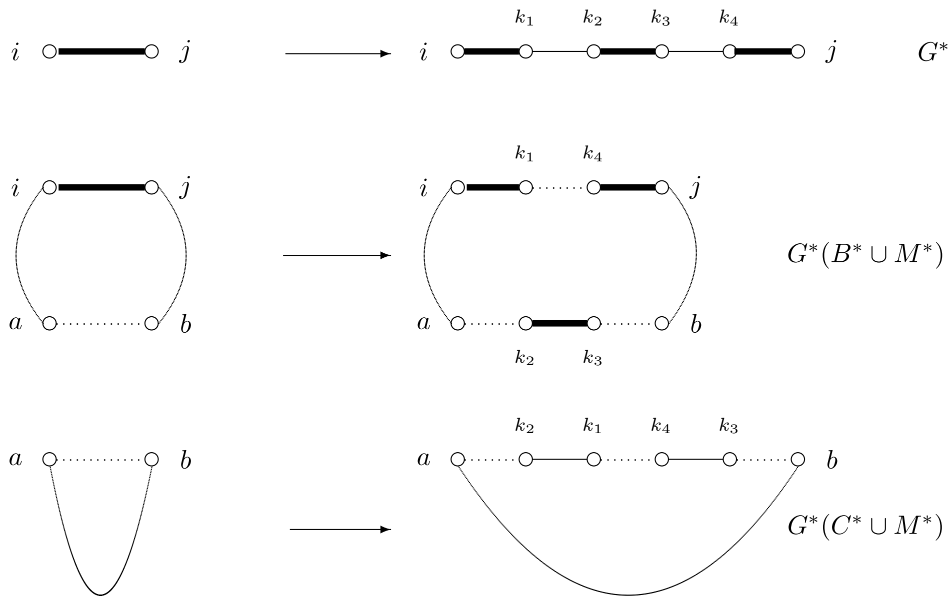


FIG. 12. *The replacement of the black edge (i, j) in procedure* DOUBLE_SUBDIVISION. *Edges in M^* are drawn as dotted lines, whereas the curved lines denote paths.*

The following lemma follows directly from the definition of G^* and Remark 1.

LEMMA 5. *The size of G^* is linear in the size of $G(\pi)$, and there is a one-to-one correspondence between alternating cycles (and alternating-cycle decompositions) of $G(\pi)$ and G^*.*

The key property of G^* and M^* is illustrated by the next lemma.

LEMMA 6. *The set M^* is a Hamiltonian matching of G^*, and every alternating cycle of G^* is oriented with respect to every permutation associated with G^* and M^*.*

Proof. First of all, at the end of each execution of the *for* loop the updated M^* is a Hamiltonian matching of the new G^*. The updating formula of M^* corresponds indeed to an insertion of the new degree-2 nodes k_1, k_2, k_3, k_4 in the previous Hamiltonian circuits $G^*(B^* \cup M^*)$ and $G^*(C^* \cup M^*)$. In $G^*(B^* \cup M^*)$, edges (i, j) and (a, b) are replaced by the paths $(i, k_1), (k_1, k_4), (k_4, j)$ and $(a, k_2), (k_2, k_3), (k_3, b)$, respectively, whereas in $G^*(C^* \cup M^*)$ edge (a, b) is replaced by the path $(a, k_2), (k_2, k_1), (k_1, k_4)$, $(k_4, k_3), (k_3, b)$; see Figure 12.

Every permutation π^* associated with G^* and M^* defines an orientation of the edges in B^*. In particular, each edge is oriented according to the direction in which it is traversed by the Hamiltonian circuit $G^*(B^* \cup M^*)$ starting from the node corresponding to π_0^* and traversing a black edge first; see section 3.

After the replacement of edge $r = (i, j)$, to every alternating cycle of $G(\pi)$ which contains this edge there corresponds an alternating cycle of G^* that contains edges $(i, k_1), (k_1, k_2), (k_2, k_3), (k_3, k_4), (k_4, j)$. Furthermore, with respect to every π^* associated with G^* and M^*, either edge (i, k_1) is oriented from i to k_1, edge (k_4, j) from k_4 to j, and edge (k_2, k_3) from k_3 to k_2 or, conversely, edge (i, k_1) is oriented from k_1 to i, edge (k_4, j) from j to k_4, and edge (k_2, k_3) from k_2 to k_3. This property is maintained throughout the procedure, since the new Hamiltonian circuits $G^*(B^* \cup M^*)$ are obtained from the previous ones by replacing edges with paths.

The above discussion shows that at the end of the procedure, when all black edges have been replaced, every alternating cycle of G^* is oriented with respect to any permutation associated with G^* and M^*. ◻

The previous lemma leads to the main result of this section.

THEOREM 9. *MIN-SBR is NP-hard.*

Proof. Consider a breakpoint graph $G(\pi)$ associated with some permutation π, and let M be the Hamiltonian matching of $G(\pi)$ corresponding to π. Construct G^* and M^* from $G(\pi)$ and M by applying procedure DOUBLE_SUBDIVISION, and let π^* be any permutation associated with G^* and M^*. By Lemmas 5 and 6 and Theorem 4, $d(\pi^*) = b(\pi^*) - c(\pi^*)$, $c(\pi) = c(\pi^*)$, and $b(\pi^*)$ is trivially determined. So one can compute the optimal value of MAX-ACD for $G(\pi)$ by solving MIN-SBR on π^*, whose size is linear in the size of $G(\pi)$. The claim then follows from Corollary 2. ◻

As one might expect, it is also easy to determine the corresponding optimal alternating-cycle decomposition of $G(\pi)$ from an optimal sequence of reversals needed to sort π^*. In particular, for each element π_i^* of π^*, consider the number of reversals in which it is involved in the sequence. Define a signed permutation $\vec{\pi}^*$ from π^* by letting element π_i^* be an even (resp., odd) element if this number is even (resp., odd). As explained in section 2, $d(\vec{\pi}^*) = d(\pi^*)$, and the unique alternating-cycle decomposition of $G(\vec{\pi}^*)$ corresponds to an alternating-cycle decomposition of $G(\pi^*)$ of cardinality $b(\vec{\pi}^*) - d(\vec{\pi}^*) = b(\pi^*) - d(\pi^*)$ and is therefore optimal. From this decomposition it is straightforward to derive an optimal alternating-cycle decomposition for $G(\pi)$.

We briefly discuss how this reduction from MAX-ACD to MIN-SBR could in principle be used to derive approximation algorithms for MAX-ACD (and therefore for MAX-ECD by Remark 2 from approximation algorithms for MIN-SBR. Let Π denote the set of all permutations and Π^* denote the set of permutations obtained from some $\pi \in \Pi$ by applying the transformation given above. Suppose one wants to approximate $c(\pi)$ for $\pi \in \Pi$ and has an approximation algorithm for MIN-SBR, which applied to a permutation $\tau \in \Pi$ delivers a solution of value $d^A(\tau)$. As described above, it is possible to derive π^* from π so that $c(\pi) = c(\pi^*) = b(\pi^*) - d(\pi^*)$. One can then compute a solution of MIN-SBR on π^* of value $d^A(\pi^*)$ and define a signed permutation $\vec{\pi}^*$ such that $d(\vec{\pi}^*) = d^A(\pi^*)$ and $b(\vec{\pi}^*) = b(\pi^*)$. (In fact, it may happen that $d(\vec{\pi}^*) < d^A(\pi^*)$, in which case the approximate solution can be improved by optimally sorting the corresponding signed permutation. We assume that the approximation algorithm includes this post-processing phase, so as to ensure $d(\vec{\pi}^*) = d^A(\pi^*)$.) The approximate solution of MAX-ACD is $c^A(\pi) = c^A(\pi^*) = c(\vec{\pi}^*) = b(\vec{\pi}^*) - d(\vec{\pi}^*) = b(\pi^*) - d^A(\pi^*)$. To guarantee that $\inf_{\pi \in \Pi} \frac{c^A(\pi)}{c(\pi)} \geq \alpha$ for some $\alpha > 0$ ($\alpha < 1$), one would need $\inf_{\pi^* \in \Pi^*} \frac{b(\pi^*) - d^A(\pi^*)}{b(\pi^*) - d(\pi^*)} \geq \alpha$; i.e.,

$$\sup_{\pi^* \in \Pi^*} \frac{d^A(\pi^*)}{d(\pi^*)} \leq \alpha + (1-\alpha)\frac{b(\pi^*)}{d(\pi^*)}.$$

Unfortunately, no approximation algorithm for general MIN-SBR known so far guarantees $\sup_{\pi \in \Pi} \frac{d^A(\pi)}{d(\pi)} \leq \alpha + (1-\alpha)\frac{b(\pi)}{d(\pi)}$. Observe that as $b(\pi) \leq 2d(\pi)$, such an algorithm should have a worst-case performance ratio of at most $2 - \alpha$.

As an immediate consequence of Theorem 9, the problem of sorting *words* by reversals (see [17]) is NP-hard. This latter problem calls for a shortest sequence of reversals transforming a string $w_1 \ \ldots \ w_n$, such that $w_i \in \{1, \ldots, m\}$ for $i = 1, \ldots, n$ and $n \geq m$, into a sorted string $y_1 \ \ldots \ y_n$, where $y_i \leq y_{i+1}$ for $i = 1, \ldots, n$ and is therefore clearly a generalization of MIN-SBR. Other relevant generalizations of MIN-SBR are mentioned in [4].

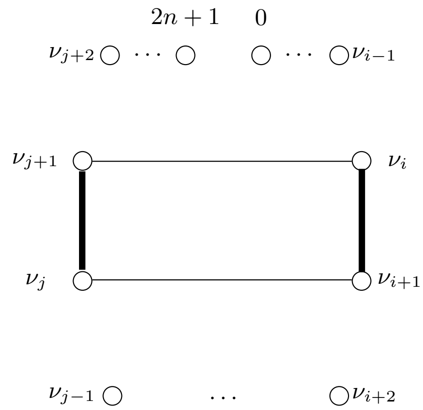


FIG. 13. *Illustration of the proof of Lemma* 7.

Our construction can easily be adapted to show that the *circular* variant of MIN-SBR (see [15]) is NP-hard as well. In this problem, reversals of intervals of the type (j, i), $j > i$, are also allowed, which transform $\pi = (\pi_1 \ \ldots \ \pi_n)$ into $(\pi_{i+j-1} \ \ldots \ \pi_{j+1} \ \pi_j \ \pi_{i+1} \ \ldots \ \pi_{j-1} \ \pi_{i+n} \ \pi_{i+n-1} \ \ldots \ \pi_{i+j})$, where all indexes are understood to be modulo n. Finally, the main ideas of the construction have been used in [4] to prove that relevant generalizations of signed MIN-SBR are NP-hard.

6. The worst-case performance of lower bound $b(\pi) - c(\pi)$ on $d(\pi)$. In this section we show that both the absolute and asymptotic worst-case performance ratio of lower bound $b(\pi) - c(\pi)$ on $d(\pi)$ are equal to $\frac{3}{2}$.

Let Π denote the set of all permutations, and Π_n the set of permutations with at least n elements. For lower bound $b(\pi) - c(\pi)$ on $d(\pi)$ we define the *absolute worst-case performance ratio* as

$$\sup_{\pi \in \Pi} \frac{d(\pi)}{b(\pi) - c(\pi)}$$

and the *asymptotic worst-case performance ratio* as

$$\lim_{n \to \infty} \sup_{\pi \in \Pi_n} \frac{d(\pi)}{b(\pi) - c(\pi)}.$$

To prove our result, we make use of some notions introduced in section 2, in particular we use the interleaving graph $H(\vec{\pi})$ of a signed permutation $\vec{\pi}$. We first prove a technical lemma. An equivalent lemma can be found in [10]; the proof is quite simple and hence is given explicitly below.

LEMMA 7. *Given a signed permutation $\vec{\pi}$, every connected component of $H(\vec{\pi})$ in which all nodes correspond to unoriented alternating cycles contains either at least two alternating cycles or one alternating cycle with at least* 6 *edges.*

Proof. Since alternating cycles have at least 4 edges, it is sufficient to prove that there cannot exist an unoriented alternating cycle with 4 edges which is not interleaving with any other alternating cycle. Suppose such an alternating cycle A exists, and let n be the number of elements of $\vec{\pi}$ and ν be the unsigned permutation corresponding to $\vec{\pi}$, as defined in section 2. As it is unoriented, A must be of the form $(\nu_i, \nu_{i+1}), (\nu_{i+1}, \nu_j), (\nu_j, \nu_{j+1}), (\nu_{j+1}, \nu_i)$, where the first and third edges are black, and the second and fourth are gray; see Figure 13. Also, $j > i + 2$ as $i + 1$ and j

are connected by a gray edge, and either $i > 0$ or $j+1 < 2n+1$ (or both), as i and $j+1$ are connected by a gray edge. The fact that A is not interleaving with any other alternating cycle means that no other gray edge in $G(\vec{\pi})$ is interleaving with (ν_{i+1}, ν_j) or (ν_{j+1}, ν_i); i.e., no gray edge connects an element ν_k, $i+2 \le k \le j-1$, to an element ν_l, $0 \le l \le i-1$ or $j+2 \le l \le 2n+1$. Therefore, as a gray edge connects ν_{i+1} to ν_j, for each element ν_k, $i+1 \le k \le j$, elements $\nu_k + 1$ and $\nu_k - 1$ are also in a position between $i+1$ and j. This leads to a contradiction. Consider the minimum and maximum elements ν_h and ν_k, $i+1 \le h, k \le j$. If $i > 0$, element $\nu_h - 1$ exists and is not in a position between $i+1$ and j. Otherwise, $j+1 < 2n+1$, hence element $\nu_k + 1$ exists and is not in a position between $i+1$ and j. □

THEOREM 10. *Both the absolute and asymptotic worst-case performance ratio of lower bound $b(\pi) - c(\pi)$ on $d(\pi)$ are equal to $\frac{3}{2}$.*

Proof. We first prove that, for any permutation $\pi \in \Pi$, $\frac{d(\pi)}{b(\pi)-c(\pi)} \le \frac{3}{2}$. Given a permutation π, suppose an optimal decomposition of $G(\pi)$ into $c(\pi)$ alternating cycles is available. As described in section 2, define from π and the given decomposition a signed permutation $\vec{\pi}$ such that $b(\vec{\pi}) = b(\pi)$, $c(\vec{\pi}) = c(\pi)$, and $d(\vec{\pi}) \ge d(\pi)$. The proof follows by showing that $\frac{d(\vec{\pi})}{b(\vec{\pi})-c(\vec{\pi})} \le \frac{3}{2}$.

From Theorem 2, $d(\vec{\pi}) = b(\vec{\pi}) - c(\vec{\pi}) + h(\vec{\pi}) + f(\vec{\pi})$; therefore all we need to show is that $\frac{h(\vec{\pi})+f(\vec{\pi})}{b(\vec{\pi})-c(\vec{\pi})} \le \frac{1}{2}$. As explained in section 2, $h(\vec{\pi}) + f(\vec{\pi})$ is at most equal to the number of connected components of the interleaving graph $H(\vec{\pi})$ in which all nodes correspond to unoriented alternating cycles. By Lemma 7, each such component contains either at least two alternating cycles or one alternating cycle with at least 6 edges. In both cases, the contribution to $b(\vec{\pi}) - c(\vec{\pi})$ of the component is at least 2, hence $\frac{h(\vec{\pi})+f(\vec{\pi})}{b(\vec{\pi})-c(\vec{\pi})} \le \frac{1}{2}$ follows.

We conclude the proof by showing a family of permutations for which the worst case is attained. Consider first the permutation

$$\pi = (0\ 5\ 6\ 3\ 4\ 1\ 2\ 7)$$

(where the dummy elements $\pi_0 = 0$ and $\pi_{n+1} = \pi_7 = 7$ have been indicated explicitly). It is easy to check that $b(\pi) = 4$, $c(\pi) = 2$, and $d(\pi) = 3$, i.e., $\frac{d(\pi)}{b(\pi)-c(\pi)} = \frac{3}{2}$, hence the absolute worst-case performance follows. Here the unique connected component of $H(\vec{\pi})$ contains two unoriented alternating cycles of 4 edges. Furthermore, for any integer $k \ge 1$, let π^{k+1} denote the permutation obtained by "duplicating" π $k+1$ times, which has the form

$$\pi^{k+1} = (0\ 5\ 6\ 3\ 4\ 1\ 2\ 7\ (0+8)\ (5+8)\ (6+8)\ (3+8)\ (4+8)\ (1+8)\ (2+8)\ (7+8)$$

$$\ldots\ (0+8k)\ (5+8k)\ (6+8k)\ (3+8k)\ (4+8k)\ (1+8k)\ (2+8k)\ (7+8k)).$$

It is easy to see that π^{k+1} is such that $b(\pi^{k+1}) = 4(k+1)$ and $c(\pi^{k+1}) = 2(k+1)$. Furthermore, one may check that the results shown by Hannenhalli and Pevzner in [11] guarantee that $d(\pi^{k+1}) = 3(k+1)$. In particular, an optimal sorting of π^{k+1} is obtained by applying to each subsequence $(0+8i)\ (5+8i)\ (6+8i)\ (3+8i)\ (4+8i)\ (1+8i)\ (2+8i)\ (7+8i)$ the reversals needed to sort π. This proves the asymptotic worst-case performance. □

One may prove the same theorem for signed permutations in an analogous way.

As a consequence of Theorem 10, by some straightforward algebraic computation one gets that the availability of an approximation algorithm for MAX-ACD with worst-case performance ratio $\alpha = \inf_{\pi \in \Pi} \frac{c^A(\pi)}{c(\pi)} < 1$ (where $c^A(\pi)$ is the value of the approximate solution for the MAX-ACD instance defined by $G(\pi)$) would yield an approximation algorithm for MIN-SBR with worst-case performance ratio $\beta = \sup_{\pi \in \Pi} \frac{d^A(\pi)}{d(\pi)} \leq \frac{3}{2} \frac{b(\pi)-\alpha c(\pi)}{b(\pi)-c(\pi)} \leq \frac{3(2-\alpha)}{2}$ (where $d^A(\pi)$ is the value of the approximate solution for the MIN-SBR instance defined by π). The last inequality follows from the obvious relation $b(\pi) \geq 2c(\pi)$. So this scheme cannot improve on the approximation ratio of $\frac{3}{2}$ due to Christie [7]. It is also worth noting that in order to prove that an approximation algorithm for MIN-SBR has a worst-case performance ratio better than $\frac{3}{2}$, it is not sufficient just to compare the value of the approximate solution with the lower bound $b(\pi) - c(\pi)$.

Acknowledgments. A preliminary version of this paper received the "Best Paper by a Young Scientist Award" in the First Annual International Conference on Computational Molecular Biology, and I would like to thank the whole conference committee, chaired by Michael Waterman, for this award. More generally, I am really grateful to the computational biology community for the enthusiastic reaction when I proved that MIN-SBR was NP-hard. Finally, I would like to thank Sridhar Hannenhalli, Jens Lagergren, Giuseppe Lancia, and Pavel Pevzner for helpful discussions on the subject, and four anonymous referees for their valuable suggestions.

REFERENCES

[1] V. Bafna and P. A. Pevzner, *Genome rearrangements and sorting by reversals*, SIAM J. Comput., 25 (1996), pp. 272–289.

[2] P. Berman and S. Hannenhalli, *Fast sorting by reversal*, in Proc. 7th Annual Symposium on Combinatorial Pattern Matching, Lecture Notes in Comput. Sci. 1075, Springer-Verlag, Berlin, 1996, pp. 168–185.

[3] A. Caprara, *Sorting Permutations by Reversals and Eulerian Cycle Decompositions*, Tech. report OR/97/4 DEIS, Università di Bologna Bologna, Italy, 1997.

[4] A. Caprara, *Formulations and Complexity of Multiple Sorting by Reversals*, in Proc. Third Annual international Conference on Computational Molecular Biology (RECOMB'99), ACM Press, New York, 1999, to appear.

[5] A. Caprara, *On the Tightness of the Alternating-Cycle Lower Bound for Sorting by Reversals*, J. Comb. Optim., to appear.

[6] A. Caprara, G. Lancia, and S.-K. Ng, *A Column-Generation Based Branch-and-Bound Algorithm for Sorting by Reversals*, Tech. report OR/95/7 DEIS, Università di Bologna, Bologna, Italy, 1995; to appear in DIMACS Series in Discrete Mathematics and Theoretical Computer Science.

[7] D. A. Christie, *A* 3/2 *approximation algorithm for sorting by reversals*, in Proc. the 9th Annual ACM-SIAM Symposium on Discrete Algorithms, ACM Press, New York, 1998, pp. 244–252.

[8] P. Crescenzi and V. Kann, *A Compendium of NP Optimization Problems*, Tech. report SI/RR-95/02, Dipartimento di Scienze dell'Informazione, Università di Roma "La Sapienza," 1995; also available online from http://www.nada.kth.se/theory/problemlist.html.

[9] N. Franklin, *Conservation of genome form but not sequence in the transcription antitermination determinants of bacteriophages λ, ϕ21 and P22*, J. Molecular Evolution, 181 (1985), pp. 75–84.

[10] S. Hannenhalli and P. A. Pevzner, *Transforming cabbage into turnip (Polynomial algorithm for sorting signed permutations by reversals)*, in Proc. 27th Annual ACM Symposium on the Theory of Computing, ACM Press, New York, 1995, pp. 178–187.

[11] S. Hannenhalli and P. A. Pevzner, *To cut ... or not to cut (Applications of comparative physical maps in molecular evolution)*, in Proc. 7th Annual ACM-SIAM Symposium on Discrete Algorithms, ACM Press, New York, 1996, pp. 304–313.

[12] I. Holyer, *The NP-completeness of some edge-partition problems*, SIAM J. Comput., 10 (1981), pp. 713–717.

[13] R. W. Irving and D. A. Christie, *Sorting by Reversals: A Conjecture of Kececioglu and Sankoff*, Computer Science Dept. Tech report TR-1995-12, University of Glasgow, Glasgow, 1995.

[14] H. Kaplan, R. Shamir, and R. E. Tarjan, *Faster and simpler algorithm for sorting signed permutations by reversals*, in Proc. 8th Annual ACM-SIAM Symposium on Discrete Algorithms, ACM Press, New York, 1997, pp. 344–351.

[15] J. Kececioglu and D. Sankoff, *Efficient bounds for oriented chromosome inversion distance*, in Proc. 5th Annual Symposium on Combinatorial Pattern Matching, Lecture Notes in Comput. Sci. 807, Springer-Verlag, New York, Berlin, 1994, pp. 307–325.

[16] J. Kececioglu and D. Sankoff, *Exact and approximation algorithms for sorting by reversals, with application to genome rearrangement*, Algorithmica, 13 (1995), pp. 180–210.

[17] P. A. Pevzner and M. S. Waterman, *Open combinatorial problems in computational molecular biology*, Proc. 3rd Israel Symposium on the Theory of Computing and Systems, IEEE Computer Society Press, Piscataway, NJ, 1995, pp. 158–172.

[18] D. Sankoff, *Analytical approaches to genomic evolution*, Biochimie, 75 (1993), pp. 409–413.

[19] D. Sankoff, G. Leduc, N. Antoine, B. Paquin, B. F. Lang, and R. Cedergren, *Gene order comparisons for phylogenetic inference: Evolution of the mitochondrial genome*, in Proc. Natl. Acad. Sci. U.S.A., 89 (1992), pp. 6575–6579.

[20] N. Tran, *An Easy Case of Sorting by Reversals*, in Proc. 8th Annual Symposium on Combinatorial Pattern Matching, Lecture Notes in Comput. Sci. 1264, Springer-Verlag, New York, Berlin, 1997, pp. 83–89.

[21] G. A. Watterson, W. J. Ewens, T. E. Hall, and A. Morgan, *The chromosome inversion problem*, J. Theoret. Biology, 99 (1982), pp. 1–7.

SIAM J. DISCRETE MATH.
Vol. 12, No. 1, pp. 111-118

MEDIAN GRAPHS AND TRIANGLE-FREE GRAPHS*

WILFRIED IMRICH†, SANDI KLAVŽAR‡, AND HENRY MARTYN MULDER§

Abstract. Let $M(m,n)$ be the complexity of checking whether a graph G with m edges and n vertices is a median graph. We show that the complexity of checking whether G is triangle-free is at most $O(M(m,m))$. Conversely, we prove that the complexity of checking whether a given graph is a median graph is at most $O(m \log n + T(m \log n, n))$, where $T(m,n)$ is the complexity of finding all triangles of the graph. We also demonstrate that, intuitively speaking, there are as many median graphs as there are triangle-free graphs. Finally, these results enable us to prove that the complexity of recognizing planar median graphs is linear.

Key words. median graph, triangle-free graph, algorithm, complexity

AMS subject classifications. 68Q25, 05C12

PII. S0895480197323494

1. Introduction. All graphs considered in this paper are finite undirected graphs without loops and multiple edges. Unless stated otherwise, for a given graph G, n and m stand for the number of its vertices and edges, respectively.

The *interval* $I(u,v)$ between vertices u and v consists of all vertices on shortest paths between u and v. A *median* of a set of three vertices u, v, and w is a vertex that lies in $I(u,v) \cap I(u,w) \cap I(v,w)$. A connected graph G is a *median graph* if every triple of its vertices has a unique median. Trees, hypercubes, and grid graphs are prime examples of median graphs. It is easy to see that median graphs are bipartite.

By now a rich theory has been developed for median graphs. For instance, they are shown to be the graphs of windex 2 by Chung, Graham, and Saks [8]. They constitute the class of retracts of hypercubes (see Bandelt [5]). They have applications in location theory and consensus theory (see, e.g., McMorris, Mulder, and Roberts [15]). They are the underlying graphs of discrete structures from various areas, involving, e.g., ternary algebras, hypergraphs, convexities, semilattices, join geometries, and conflict models. For a survey of all these aspects of median graphs, the reader is referred to Klavžar and Mulder [14].

It is clear that median graphs can be recognized in polynomial time and a direct approach would yield an $O(n^4)$ algorithm. Jha and Slutzki [13] followed the convex expansion theorem of Mulder [16, 17] to obtain an $O(mn) = O(n^2 \log n)$ algorithm. A simple algorithm of the same complexity was recently proposed by Imrich and Klavžar [11]. Currently, the fastest known algorithm for recognizing median graphs is by Hagauer, Imrich, and Klavžar [9] and runs in $O(m\sqrt{n}) = O(n^{3/2} \log n)$ time. The

*Received by the editors June 25, 1997; accepted for publication (in revised form) July 22, 1998; published electronically January 29, 1999.

http://www.siam.org/journals/sidma/12-1/32349.html

†Department of Mathematics and Applied Geometry, Montanuniversität Leoben, A-8700 Leoben, Austria (imrich@unileben.ac.at).

‡Department of Mathematics, PEF, University of Maribor, Koroška cesta 160, 2000 Maribor, Slovenia (sandi.klavzar@uni-lj.si). The work of this author was supported by the Ministry of Science and Technology of Slovenia under grant J1-0498-0101.

§Econometrisch Instituut, Erasmus Universiteit, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands (martyn@wis.few.eur.nl). The work of this author was supported by the Ministry of Science and Technology of Slovenia and by the NUFFIC, the Netherlands.

last equality holds because a median graph with n vertices has at most $n \log n$ edges. For more information on these and related algorithms, see [10].

Cartesian product graphs can be recognized in $O(m \log n)$ time by the algorithm of Aurenhammer, Hagauer, and Imrich [3]. The simplest Cartesian product graphs are obtained by multiplying complete graphs on two vertices and are usually called hypercubes or n-cubes. As mentioned above, Bandelt [5] proved that median graphs are very special subgraphs of n-cubes; namely, they are precisely the retracts of the n-cubes. Hence, the natural question arises: Can the complexity $O(m\sqrt{n})$ for recognizing median graphs be improved to, say, $O(m \log^k n)$ for some $k \geq 1$? The main message of this paper claims that this is very unlikely.

Several algorithms are known which recognize triangle-free graphs or, more generally, find all triangles of a given graph. Clearly, a straightforward implementation yields an algorithm of complexity $O(mn)$. It is worthwhile to add that this simple algorithm finds a triangle in $O(n^{5/3})$ on the average (see [12]). In [12] Itai and Rodeh show that Strassen's algorithm for (Boolean) matrix multiplication can be used to solve the problem in $O(n^{\log 7})$ time. In addition, they also give an algorithm using rooted spanning trees of complexity $O(m^{3/2})$. The algorithm finds all triangles of a given graph and becomes linear in the case of planar graphs. Another algorithm which lists all the triangles of a given graph is due to Chiba and Nishizeki [7]. For a graph G its time complexity is $O(a(G)m)$, where $a(G)$ denotes the arboricity of G. They also show that $a(G) \leq O(m^{1/2})$. Thus, the algorithm of Chiba and Nishizeki is in the worst case still of complexity $O(m^{3/2})$. Recently, Alon, Yuster, and Zwick [2] proved that deciding whether a directed or undirected graph contains a triangle, and finding one if it does, can be done in $O(m^{1.41})$ time. For related results we refer to [1, 18].

We continue this paper as follows. We first recall several notions needed in the rest of the paper. Then, in the next section, we introduce the main construction of this paper which for a given triangle-free graph produces a median graph. We study this construction and show that it can be used to deduce that, intuitively speaking, there are as many median graphs as there are triangle-free graphs. In section 3 we use the construction to show that recognizing triangle-free graphs is at most as difficult as recognizing median graphs. For the converse we prove that the complexity of checking whether a graph with m edges and n vertices is a median graph is at most $O(m \log n + T(m \log n, n))$, where $T(m, n)$ is the complexity of finding all triangles of a given graph with m edges and n vertices. A consequence of this relationship is a linear algorithm for the recognition of planar median graphs. It exploits the fact that the triangles of a planar graph can be found in linear time.

The *eccentricity* $e(x)$ of a vertex x in a connected graph G is the maximum distance of x to any other vertex in G. The *radius* $r(G)$ of G is the minimum eccentricity in G, and a vertex x is a *central vertex* of G if $e(x) = r(G)$. The *periphery* of G consists of all vertices in G at distance $r(G)$ to some central vertex in G.

For an edge $e = uv$ in a graph G, the *subdivision* of e is obtained by replacing the edge e by a new vertex adjacent to both u and v. For convenience, we denote the new vertex by e and the new edges by ue and ev.

Let G be a graph. The *simplex graph* $S(G)$ of G is the covering graph of the partially ordered set of the family of simplices (i.e., complete subgraphs) in G ordered by inclusion. In other words, the vertices of $S(G)$ are the complete subgraphs of G (including the empty one), two vertices being adjacent provided they differ in at most one vertex. Simplex graphs were introduced by Bandelt and van de Vel [6].

Obviously, a simplex graph is a median graph: The median of the simplices A, B, C is the simplex $(A \cap B) \cup (A \cap C) \cup (B \cap C)$.

By Q_3^- we denote the graph obtained from the 3-cube Q_3 by deleting one vertex. The antipodal of the deleted vertex is called the *base* of the Q_3^-. In other words, the base is the only vertex of Q_3^- which is incident to three vertices of degree 3. Note that the three vertices of degree 2 in Q_3^- do not have a median.

Finally, the *Cartesian product* $G \square H$ of graphs G and H is the graph with vertex set $V(G) \times V(H)$ and $(a, x)(b, y) \in E(G \square H)$ whenever $ab \in E(G)$ and $x = y$, or $a = b$ and $xy \in E(H)$.

2. Constructing median graphs from triangle-free graphs. Let $G = (V, E)$ be a graph with $|V| = n$ and $|E| = m$. The graph $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ is obtained from G by subdividing all edges of G and adding a new vertex z joined to all the original vertices of G. So we have $\widetilde{V} = V \cup E \cup \{z\}$ and

$$\widetilde{E} = \{zv \mid v \in V\} \cup \{ue \mid e \in E, u \in V \text{ and } u \text{ is incident with } e \text{ in } G\}.$$

Note that $|\widetilde{V}| = n + m + 1$, and $|\widetilde{E}| = n + 2m$. Observe also that $\widetilde{G}$ is connected, even if G is not. An example for this construction is given in Fig. 2.1.

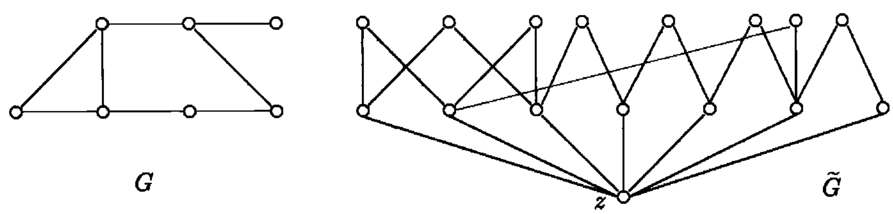


FIG. 2.1. *Illustration of the construction.*

Let d denote the degree function of G and $\widetilde{d}$ that of $\widetilde{G}$. Then we have $\widetilde{d}(z) = n$, and $\widetilde{d}(v) = d(v) + 1$, for $v \in V$, and $\widetilde{d}(e) = 2$, for $e \in E$. Note that z has maximum degree in $\widetilde{G}$, and that $\widetilde{d}(v) = n$ if and only if v is a *dominating vertex* in G, i.e., a vertex adjacent to all other vertices in G.

Since all vertices in $\widetilde{G}$ are at a distance of at most 2 from z, we have $r(\widetilde{G}) \leq 2$. Clearly, we have $r(\widetilde{G}) = 2$ if and only if $m \geq 1$. In any case, z is a central vertex of $\widetilde{G}$. Note that G is disconnected if and only if z is a cut-vertex in $\widetilde{G}$, so that $\widetilde{G}$ is 2-connected if and only if G is nontrivial and connected.

Assume that there is an edge $e = uv$ and a vertex w in G such that w is not incident with e. Then, in $\widetilde{G}$, the vertices w and e have distance 3, so neither vertex is central in $\widetilde{G}$. This implies the following facts: Either

(i) $G = K_2$, and $\widetilde{G} = C_4 = K_2 \square K_2$, and all four vertices of $\widetilde{G}$ are central, or

(ii) $G = K_{1,n-1}$ with $n \neq 2$, and $\widetilde{G} = K_2 \square K_{1,n-1}$, and the two vertices of degree n in $\widetilde{G}$ are the two central vertices of $\widetilde{G}$ (where it is understood that $K_{1,0} = K_1$), or

(iii) G is not a star and z is the unique central vertex of $\widetilde{G}$.

Thus, to reconstruct G from $\widetilde{G}$, we need only to search for a central vertex; take this to be z, take the neighbors of z to be the vertices of G, and replace the remaining vertices, which are all of degree 2, by edges.

An automorphism of a graph necessarily maps central vertices to central vertices. So, if $G = (V, E)$ is not a star, then each automorphism α of $\widetilde{G}$ fixes z. Furthermore, we have $\alpha(V) = V$ and $\alpha(E) = E$ in $\widetilde{G}$. So, essentially, $\alpha|_{V \cup E}$ is an automorphism of G, which gives us the following result.

PROPOSITION 2.1. *Let G be a graph. If $G = K_2$, then $Aut(\widetilde{G}) = Aut(C_4)$. If G is a star different from K_2, then $Aut(\widetilde{G}) \cong \mathbb{Z}_2 \times Aut(G)$. If G is not a star, then $Aut(\widetilde{G}) \cong Aut(G)$.*

Note that G contains a triangle if and only if $\widetilde{G}$ contains a Q_3^-, the base of which necessarily is z.

In case that G is triangle-free, the graph $\widetilde{G}$ is just the simplex graph of G and hence is a median graph. Since Q_3^- is a forbidden convex subgraph in a median graph, we have as an immediate consequence the following result.

THEOREM 2.2. *A graph G is triangle-free if and only if its associated graph $\widetilde{G}$ is a median graph.*

We next have a closer look at the median graphs arising in Theorem 2.2.

Let G be a triangle-free graph. Then G contains a dominating vertex if and only if G is a star. Hence, if G is not a star, then z is not only the unique central vertex but also the unique vertex of maximum degree in $\widetilde{G}$. The only vertices of degree 1 in $\widetilde{G}$ arise from components of G consisting of a single vertex. Let us ignore such components. Then the minimum degree in $\widetilde{G}$ is 2.

Conversely, let H be a median graph of minimum degree 2, with radius $r(H) = 2$, with a unique central vertex z, which is also the unique vertex of maximum degree n. Let p be any vertex in the periphery of H, whence at distance 2 from z. Since H is bipartite, all neighbors of p must be adjacent to z. Since H is median, and therefore without $K_{2,3}$, it follows that p has exactly two neighbors. Let m be the number of vertices in the periphery of H. Then H has $1 + n + m$ vertices and $n + 2m$ edges. Now we construct the graph G on the set of neighbors of z in H. We join two vertices of G by an edge if and only if in H they have a common neighbor in the periphery. Then G has n vertices and m edges and, clearly, we have $H = \widetilde{G}$. Since H is Q_3^--free, G is triangle-free.

Let $\mathcal{G}_{n,m}$ be the class of triangle-free graphs with n vertices and m edges and without singleton components. Let $\mathcal{H}_{n,m}$ be the class of median graphs with minimum degree 2 and radius 2 and a single vertex of maximum degree n, which is also the unique central vertex, and m vertices in the periphery. Thus, we have just proved the following theorem.

THEOREM 2.3. *For each n and m the mapping $G \mapsto \widetilde{G}$ is a bijection between the graph classes $\mathcal{G}_{n,m}$ and $\mathcal{H}_{n,m}$.*

Thus we have an injection of the class $\mathcal{T}$ of triangle-free graphs into the class $\mathcal{M}_2$ of median graphs of radius 2. Let $\mathcal{M}^*$ be the class of all Q_3-free median graphs and let $\mathcal{M}$ be the class of all median graphs. Then we have the following situation:

$$\mathcal{M} \subset \mathcal{T} \hookrightarrow \mathcal{M}_2 \subset \mathcal{M}^* \subset \mathcal{M}\,.$$

Intuitively speaking, we have shown that there are as many median graphs as there are triangle-free graphs. Thus median graphs are much less exotic than one would expect from the definition of median graphs and the rich structure theory by now developed for median graphs.

We conclude this section with the following observation. Let G be a triangle-free graph. Then $\widetilde{G}$ is a median graph and can be isometrically embedded into a hypercube Q_r. Let $i(\widetilde{G})$ be such an embedding. Let uv be an edge of G. Then the corresponding vertices $\widetilde{u}$ and $\widetilde{v}$ lie on a 4-cycle of $\widetilde{G}$. Since i is an isometry, it maps a 4-cycle of $\widetilde{G}$ onto a 4-cycle of Q_r. This in particular implies that $d(i(\widetilde{u}), i(\widetilde{u})) = 2$. Hence we have the following proposition.

PROPOSITION 2.4. *Let G be a triangle-free graph. Then there is an r and a mapping $j : V(G) \to V(Q_r)$, such that if uv is an edge of G, then $d(j(u), j(v)) = 2$.*

3. On the complexity of recognizing median graphs and triangle-free graphs. In this section we will show that the complexity of recognizing median graphs is closely related to the complexity of recognizing triangle-free graphs and to the complexity of finding all triangles of a graph. We first have the following corollary to Theorem 2.2.

COROLLARY 3.1. *Let $M(m,n)$ be the complexity of checking whether a graph G with m edges and n vertices is median. Then the complexity of checking whether G is triangle-free is at most $O(M(m,m))$.*

Proof. By Theorem 2.2 a graph G is triangle-free if and only if $\widetilde{G}$ is a median graph. Since $|E(\widetilde{G})| = 2m + n$ and $|V(\widetilde{G})| = n + m + 1$, $\widetilde{G}$ can be checked if it is a median graph with complexity $O(M(2m+n, n+m+1)) = O(M(m,m))$. ◻

We can now explain why it seems unlikely that the complexity $O(m\sqrt{n})$ for recognizing median graphs can be improved to $O(m \log^k n)$ for some $k \geq 1$. For, if this were the case, then Corollary 3.1 would imply the existence of an algorithm for recognizing triangle-free graphs of time complexity in $O(m \log^k m) = O(m \log^k n)$, thus significantly improving known algorithms for recognizing triangle-free graphs. Note also that, by Corollary 3.1, the fastest known algorithm for recognizing median graphs, which is of complexity $O(m\sqrt{n})$, yields an $O(m^{3/2})$ algorithm for recognizing triangle-free graphs.

We next consider whether algorithms for recognizing triangle-free graphs can help us in recognizing median graphs, in particular by improving the performance of the algorithm of Hagauer, Imrich, and Klavžar [9] of complexity $O(m\sqrt{n})$. As this algorithm is rather involved, we shall not recall it here in detail but will state whatever is needed for our construction. We refer to this algorithm as Algorithm A.

First some notation. Let $G = (V, E)$ be a connected bipartite graph. For $u \in V(G)$, let $N(u)$ be the set of all vertices adjacent to u. For $X \subseteq V(G)$, let $\langle X \rangle$ denote the subgraph induced by X. A subgraph H of a graph G is an *isometric* subgraph, if the distance in G between any pair of vertices u and v of H is equal to the distance between u and v in H. For any edge ab of G, we write

$$\begin{aligned}
W_{ab} &= \{w \in V \mid d(w,a) < d(w,b)\},\\
W_{ba} &= \{w \in V \mid d(w,b) < d(w,a)\},\\
U_{ab} &= \{u \in W_{ab} \mid u \text{ is adjacent to a vertex in } W_{ba}\},\\
U_{ba} &= \{u \in W_{ba} \mid u \text{ is adjacent to a vertex in } W_{ab}\},\\
F_{ab} &= \{uv \mid u \in U_{ab}, v \in U_{ba}\}.
\end{aligned}$$

We refer to the set $F = F_{ab}$ as a color. In fact, if G is a median graph, then the sets of type F are a proper edge-coloring of G. Also, G is a median graph if and only if, for any edge ab, the sets U_{ab} and U_{ba} are convex. This characterization was proved by Bandelt [4] but also follows immediately from results in [16, 17]. The bottleneck in testing whether G is a median graph is testing whether the sets U_{ab} are convex. This convexity testing can be reduced to testing condition (iii) listed below. In fact, with one exception, all steps of Algorithm A require at most $O(m \log n)$ time, the exception being Step 3.4, which tests condition (iii) for U_{ab}.

THEOREM 3.2. *Let $G = (V, E)$ be a connected bipartite graph, and let $ab \in E$. Suppose the following properties hold:*

(i) *F_{ab} is a matching that defines an isomorphism between $\langle U_{ab} \rangle$ and $\langle U_{ba} \rangle$;*

(ii) *for any $u \in U_{ab}$ and $v \in U_{ba}$, $I(u, a) \subseteq U_{ab}$ and $I(v, b) \subseteq U_{ba}$, respectively;*

(iii) *for any $u \in W_{ab} \backslash U_{ab}$ and $v \in W_{ba} \backslash U_{ba}$, $|N(u) \cap U_{ab}| \leq 1$ and $|N(v) \cap U_{ba}| \leq 1$.*

Then G is a median graph if and only if $\langle W_{ab} \rangle$ and $\langle W_{ba} \rangle$ are median graphs.

As we mentioned, Algorithm A without Step 3.4 checks all conditions of Theorem 3.2 except (iii) for U_{ab}. We first describe how Algorithm A checks this condition. It first constructs a breadth first search tree, say T, with root a. Suppose that a vertex x from $W_{ab} \setminus U_{ab}$ has two neighbors in U_{ab}, say u and v. As U_{ab} is isometric, there is a vertex $w \in U_{ab}$ which is adjacent to both u and v. Moreover, because condition (i) was also tested before, there are vertices u', v', and w' in U_{ba} which are adjacent to u, v, and w, respectively, such that these six vertices together with x induce a Q_3^-.

Let $L_0, L_1, \ldots$ be the distance levels of the tree T and assume that $x \in L_{i+1}$. Then we know that u and v both belong to L_i as condition (ii) of Theorem 3.2 has already been tested at this stage. Suppose that $w \in L_{i+1}$. Then by the down-closure there is a vertex $r \in L_{i-1}$ adjacent to u and v. But then the vertices x, w, u, v, r induce a $K_{2,3}$, which has been tested before. Hence $w \in L_{i-1}$. We thus have the situation depicted in Fig. 3.1.

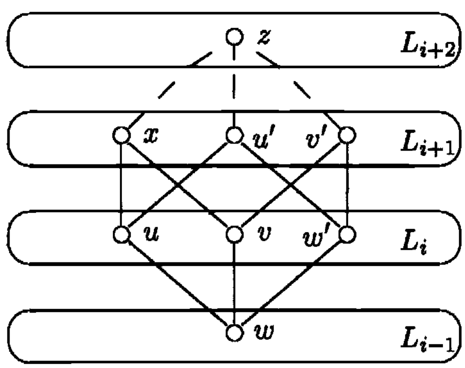


FIG. 3.1. *Testing condition* (iii).

What we need to check now is if there exists a vertex $z \in L_{i+2}$ adjacent to x, u', and v'. If this is not the case, then the test of (iii) fails and G is not a median graph. If no such situation occurs, then all the checks of Theorem 3.2 have been done and G is recognized as a median graph. We next describe how we can do these tests using an algorithm for listing all triangles of a given graph.

Let H_i be the graph on the vertex set L_i and two vertices of H_i are adjacent, if they have a common neighbor in L_{i+1}. By Corollary 4.2 of [9], H_i has at most $|L_{i+1}| \log^2 n$ edges. Thus, all the graphs H_i together have at most $n \log^2 n$ edges. Moreover, there are at most $n \log^3$ triangles in them. We now use an algorithm which finds all triangles in the graphs H_i. For each such triangle of H_i we have only to check whether the corresponding three vertices in L_{i+1} have a common neighbor z in L_{i+2}. This is easy, because G has already been embedded into a hypercube by the previous steps of Algorithm A. In other words, we know precisely the colors of the possible edges between z and the three vertices of L_{i+1}.

Suppose that we have an algorithm of complexity $T(m, n)$ which finds all triangles in a given graph with n vertices and m edges. As we wish to test whether a given graph G is a median graph, we have $m = O(n \log n)$. As already mentioned, all steps of Algorithm A, except Step 3.4, require $O(m \log n)$ time. For the test of (iii) we then follow the above approach, which takes $O(n \log^2 n, n)$ time. Thus, we have proved the following theorem.

THEOREM 3.3. *Let $T(m, n)$ be the complexity of finding all triangles of a given graph with m edges and n vertices. Then the complexity of checking whether a graph G on n vertices and m edges is a median graph is at most $O(m \log n + T(m \log n, n))$.*

As we mentioned in the introduction, the best general algorithm known for listing all triangles of a given graph is of complexity $O(m^{3/2})$. Thus, applying Theorem 3.3, we conclude that median graphs can be recognized in $O(m \log n + (m \log n)^{3/2})$ time. Since $m = O(n \log n)$ this reduces to $O(n^{3/2} \log^3 n)$ which differs only by factor $\log^2 n$ from the complexity of Algorithm A. In special cases this complexity can be further reduced. As an example we show that planar median graphs can be recognized in linear time.

The arguments leading to this result rely on the observation that the factor $\log n$ in Algorithm A without Step 3.4 is a bound on the down-degree of vertices in an isometric subgraph of the hypercube with respect to a distance tree. To be more precise, let x be vertex of in level L_{i+1} with respect to a distance tree of a graph G. Then the number of neighbors of x in L_i is at most $\log n$ if G is a subgraph of a hypercube. This number is called the down-degree of x. See [9].

In [9] it is also shown that every vertex x of down-degree k in a median graph G is contained in a hypercube Q_k. Since Q_k is nonplanar for $k > 3$ this implies that the down-degrees of planar median graphs are bounded by 3 and that Algorithm A without Step 3.4 can be executed in $O(m)$ steps.

COROLLARY 3.4. *Planar median graphs can be recognized in linear time.*

Proof. Let G be a graph on n vertices with m edges. We wish to show that the complexity of checking whether G is a planar median graph is $O(m+n)$. As it is well known that connectedness, bipartiteness, and planarity can be checked in linear time, we can assume that G is a connected, planar, bipartite graph given by its adjacency list and that we wish to check whether it is a median graph. We further observe that a distance-tree can be found in linear time and that down-degrees can be found and checked in linear time, too.

We now consider an embedding of G in the plane and the subgraph X_i spanned by L_{i+1} and L_i. In L_{i+1} there may be vertices of degree 1, 2, or 3 in X_i. Let w be a vertex in L_{i+1} of degree 3 and a, b, c be its neighbors in L_i. We split w into three vertices x, y, z and replace the edges aw, bw, cw with ax, ay, by, bz, cz, cx. We do this for every vertex of degree 3. Clearly the new graph X_i' obtained this way is still planar. Moreover, every vertex of X_i not in L_i has degree 1 or 2. We now delete the vertices of degree 1 and replace every path $x_1x_2x_3$, where $x_1, x_3 \in L_i$ and $x_2 \in L_{i+1}$, by a single edge x_1x_3. This way we obtain the graph H_i from the construction in the proof of Theorem 3.3.

Proceeding as in the proof of Theorem 3.3 we have to find the triangles in the H_i and perform certain checks, the complexity of these operations being determined by the complexity of finding all triangles. Now, the triangles in planar graphs can be found in linear time; cf. [7]. Now the proof is completed by the observation that the total number of edges in the H_i is at most $3n$, where n is the number of vertices of G. ∎

4. Concluding remark. A variant of Theorem 3.5 from [2] can be used to further improve the recognition complexity of median graphs from $O(n^{1.5} \log n)$ to $O(n^{1.41} \log^{2.82} n)$. This will be subject of a subsequent paper.

REFERENCES

[1] N. Alon, R. Yuster, and U. Zwick, *Color-coding*, J. Assoc. Comput. Mach., 42 (1995), pp. 844–856.

[2] N. Alon, R. Yuster, and U. Zwick, *Finding and counting given length cycles*, Algorithmica, 17 (1997), pp. 209–223.

[3] F. Aurenhammer, J. Hagauer, and W. Imrich, *Cartesian graph factorization at logarithmic cost per edge*, Comput. Complexity, 2 (1992), pp. 331–349.

[4] H.-J. Bandelt, *Characterizing Median Graphs*, manuscript, 1982.

[5] H.-J. Bandelt, *Retracts of hypercubes*, J. Graph Theory, 8 (1984), pp. 501–510.

[6] H.-J. Bandelt and M. van de Vel, *Superextensions and the depth of median graphs*, J. Combin. Theory Ser. A, 57 (1991), pp. 187–202.

[7] N. Chiba and T. Nishizeki, *Arboricity and subgraph listing algorithms*, SIAM J. Comput., 14 (1985), pp. 210–223.

[8] F.R.K. Chung, R.L. Graham, and M.E. Saks, *Dynamic search in graphs*, in Discrete Algorithms and Complexity, H. Wilf, ed., Academic Press, New York, 1987, pp. 351–387.

[9] J. Hagauer, W. Imrich, and S. Klavžar, *Recognizing median graphs in subquadratic time*, Theoret. Comput. Sci., to appear.

[10] W. Imrich and S. Klavžar, *On the complexity of recognizing Hamming graphs and related classes of graphs*, European J. Combin., 17 (1996), pp. 209–221.

[11] W. Imrich and S. Klavžar, *A convexity lemma and expansion procedures for bipartite graphs*, European J. Combin., 19 (1998), pp. 677–686.

[12] A. Itai and M. Rodeh, *Finding a minimum circuit in a graph*, SIAM J. Comput., 7 (1978), pp. 413–423.

[13] P.K. Jha and G. Slutzki, *Convex-expansion algorithms for recognizing and isometric embedding of median graphs*, Ars Combin., 34 (1992), pp. 75–92.

[14] S. Klavžar and H.M. Mulder, *Median graphs: Characterizations, location theory and related structures*, J. Combin. Math. Combin. Comput., to appear.

[15] F.R. McMorris, H.M. Mulder, and F.S. Roberts, *The median procedure on median graphs*, Discrete Appl. Math., 84 (1998), pp. 165–181.

[16] H.M. Mulder, *The structure of median graphs*, Discrete Math., 24 (1978), pp. 197–204.

[17] H.M. Mulder, *The Interval Function of a Graph*, Mathematical Centre Tracts 132, Mathematisch Centrum, Amsterdam, 1980.

[18] R. Yuster and U. Zwick, *Finding even cycles even faster*, SIAM J. Discrete Math., 10 (1997), pp. 209–222.

SIAM J. DISCRETE MATH.
Vol. 12, No. 1, pp. 119–135

CONVEXITY AND HHD-FREE GRAPHS*

FEODOR F. DRAGAN†§, FALK NICOLAI‡, AND ANDREAS BRANDSTÄDT†

Abstract. It is well known that chordal graphs can be characterized via m-convexity. In this paper we introduce the notion of m^3-convexity (a relaxation of m-convexity) which is closely related to semisimplicial orderings of graphs. We present new characterizations of HHD-free graphs via m^3-convexity and obtain some results known from [B. Jamison and S. Olariu, Adv. Appl. Math., 9 (1988), pp. 364–376] as corollaries. Moreover, we characterize weak bipolarizable graphs as the graphs for which the family of all m^3-convex sets is a convex geometry. As an application of our results we present a simple efficient criterion for deciding whether a HHD-free graph contains a r-dominating clique with respect to a given vertex radius function r.

Key words. convexity, convex geometry, antimatroid, chordal graphs, HHD-free graphs, weak bipolarizable graphs, semisimplicial ordering, lexicographic breadth first search, dominating clique problem

AMS subject classifications. 05C65, 05C75, 68R10

PII. S0895480195321718

1. Introduction. This paper was inspired by the results of Farber and Jamison [16] on convexity in chordal graphs and by the results of Jamison and Olariu [19] on semisimplicial orderings of graphs produced by "lexicographic breadth first search" (LexBFS) [25] and "maximum cardinality search" (MCS) [28].

Throughout this paper all graphs $G = (V, E)$ are finite, undirected, and simple (i.e., loop-free and without multiple edges). The *complement* of a graph G is the graph $\overline{G}$ with the same vertex set V, where two vertices are adjacent in $\overline{G}$ iff they are nonadjacent in G.

A *path* is a sequence of vertices $v_0, \ldots, v_l$ such that $v_i v_{i+1} \in E$ for $i = 0, \ldots, l-1$; its *length* is l. An *induced path* is a path, where $v_i v_j \in E$ iff $i = j-1$ and $j = 1, \ldots, l$. An *induced cycle* is a sequence of vertices $v_0, \ldots, v_k$ such that $v_0 = v_k$ and $v_i v_j \in E$ iff $|i - j| = 1$ (modulo k). The *length* $|C|$ of a cycle C is its number of vertices. Let also $|P|$ be the number of vertices of a path P. A *hole* is an induced cycle of length at least five, whereas an *antihole* is the complement of a hole. By P_k we denote an induced path on k vertices. A graph G is *connected* iff for any pair of vertices of G there is a path in G joining these vertices. A set $S \subset V$ is *connected* in G iff the subgraph $G(S)$ induced by S is connected.

The *distance* $d_G(u, v)$ between two vertices u, v is the minimum number of edges on a path connecting these vertices, and is infinite if u and v lie in distinct connected components of the graph G. If no confusion can arise we will omit the index G. For

*Received by the editors June 14, 1995; accepted for publication (in revised form) February 24, 1998; published electronically January 29, 1999. Results of this paper were presented at the 5th Internat. Colloq. on "Graphs and Combinatories," Marseille Luminy, September 3–8, 1995.

http://www.siam.org/journals/sidma/12-1/32171.html

†Fachbereich Informatik, Universität Rostock, Lehrstuhl für Theoretische Informatik, D 18051 Rostock, Germany (dragan@informatik.uni-rostok.de, ab@informatik.uni-rostock.de). The research of the first author was supported by DAAD

‡Gerhard-Mercator-Universität, GH Duisburg, FB Mathematik, FG Informatik, D 47048 Duisburg, Germany (nicolai@informatik.uni-duisburg.de). The research of this author was supported by DFG.

§This author is on leave from the Universitatea de stat din Moldova, Chişinău.

a vertex $v \in V$ and a set $S \subseteq V$ we denote by $d(v, S)$ the minimum over all distances $d(v, s)$, $s \in S$. A subgraph H of a graph G is *isometric* iff the distance between any pair of vertices in H is the same as that in G.

The kth *neighborhood* $N^k(v)$ of a vertex v of G is the set of all vertices of distance k to v, i.e.,

$$N^k(v) := \{u \in V : d_G(u, v) = k\},$$

whereas the *disk* of radius k centered at v is the set of all vertices of distance at most k to v:

$$D_G(v, k) := \{u \in V : d_G(u, v) \leq k\}.$$

Again, if no confusion arises we will omit the index G. We also write $N(v)$ instead of $N^1(v)$.

The *eccentricity* $e(v)$ of a vertex $v \in V$ is the maximum value of $d(v, x)$ taken over all vertices $x \in V$. The *radius* $rad(G)$ of G is the minimum eccentricity of a vertex of G, whereas the *diameter* $diam(G)$ of G is the maximum eccentricity of a vertex of G.

Now we will give a short introduction to the theory of convex geometry related to graph theory following [16] (for more information on abstract convexity and antimatroids the interested reader can consult [21]). Let V be a finite set and $\mathfrak{M}$ be a family of subsets of V. $\mathfrak{M}$ is called *alignment* of V iff the family $\mathfrak{M}$ is closed under intersection and contains both V and the empty set. Elements of $\mathfrak{M}$ will be considered as convex sets. An *aligned space* is a pair $(V, \mathfrak{M})$, where $\mathfrak{M}$ is an alignment of V.

The smallest member of $\mathfrak{M}$ containing a given set $S \subseteq V$ is the *hull* of S, denoted by $\mathfrak{M}(S)$. An element x of a set $X \in \mathfrak{M}$ is an *extreme point* of X iff $X \setminus \{x\} \in \mathfrak{M}$.

The Caratheodory number of an aligned space $(V, \mathfrak{M})$ is the minimum integer k such that for all $X \subseteq V$, $\mathfrak{M}(X)$ is the union of the hulls of all subsets Y of X such that $|Y| \leq k$.

A *convex geometry (antimatroid)* on a finite set is an aligned space satisfying the following additional property.

Minkowski–Krein–Milman property. Every convex set is the hull of its extreme points.

Equivalently, a convex geometry is an aligned space satisfying the following property.

Antiexchange property. For any convex set S and two distinct points $x, y \notin S$, $x \in \mathfrak{M}(S \cup \{y\})$ implies $y \notin \mathfrak{M}(S \cup \{x\})$.

For any convex geometry the following fundamental result holds.

THEOREM 1.1 (see [16]). *If $(V, \mathfrak{M})$ is a convex geometry, then $S \in \mathfrak{M}$ iff is an ordering $(x_1, \ldots, x_k)$ of $V \setminus S$ such that x_i is an extreme point of $S \cup \{x_i, \ldots, x_k\}$ for each $i = 1, \ldots, k$.*

For a given ordering $(v_1, \ldots, v_n)$ of the vertex set of a graph $G = (V, E)$ let $G_i := G(\{v_i, \ldots, v_n\})$ be the subgraph of G induced by the set $\{v_i, \ldots, v_n\}$, $i = 1, \ldots, n$.

Numerous classes of graphs can be characterized in the following way. G is a member of class $\mathfrak{G}$ iff there is an ordering $(v_1, \ldots, v_n)$ of $V(G)$ such that v_i satisfies a certain property $\mathbf{P}$ in the subgraph G_i, $i = 1, \ldots, n$.

Theorem 1.1 suggests that such classes of graphs might be related to convex geometries, and so it is natural to ask for a graph theoretical description of convex sets of this aligned space. On the other hand, given a collection $\mathfrak{M}$ of subsets of $V(G)$, one can ask when $(V(G), \mathfrak{M})$ is a convex geometry.

For example, if property **P** means "is simplicial" then $\mathfrak{G}$ is the class of *chordal* graphs, i.e., the graphs without induced cycles of length at least four [7, 24]. A vertex v of G is called *simplicial* iff $D(v,1)$ induces a complete subgraph of G, and *nonsimplicial* otherwise. It is well known that a graph is chordal iff it has a *perfect elimination ordering*, i.e., an ordering $(v_1,\dots,v_n)$ of V such that v_i is simplicial in G_i for each $i=1,\dots,n$ (cf. [7, 24]). Moreover, there are two linear time algorithms for computing perfect elimination orderings of chordal graphs: LexBFS [25] and MCS [28].

Two types of convexity in graphs have been studied most extensively, namely, monophonic (m-) convexity and geodesic (g-) convexity (see, e.g., [4, 12, 13, 14, 15, 16, 17, 20, 22, 26, 27]). A set $S \subseteq V(G)$ is *m-convex* (*g-convex*) iff S contains every vertex on every induced (shortest) path between vertices in S. Both types of convexity have a relation to simplicial vertices; a vertex v is an extreme point of a m-convex (g-convex) set S iff v is simplicial in $G(S)$. In [16] it is shown that G is a chordal graph iff the monophonic alignment of G is a convex geometry, while the geodesic alignment of G is a convex geometry iff G is a chordal graph without induced 3-fan (i.e., a P_4 with an additional vertex adjacent to all vertices of P_4). To prove that the monophonic alignment of a chordal graph is a convex geometry, the authors of [16] show the following nice result. Every nonsimplicial vertex of a chordal graph lies on an induced path between simplicial vertices.

For any notion of convexity on the vertex set of G, at least four degrees of local convexity may be distinguished [17]:

(1.1) $D(v,1)$ is convex for every vertex v of G,

(1.2) $D(v,k)$ is convex for every vertex v of G and every $k \geq 1$,

(1.3) $\bigcup_{v\in S} D(v,1)$ is convex for every convex subset $S \subseteq V$ of G,

(1.4) $\bigcup_{v\in S} D(v,k)$ is convex for every convex subset $S \subseteq V$ of G and every $k \geq 1$.

In [16] it was shown that for m-convexity the conditions (1.1)–(1.4) are equivalent and hold iff the graph is chordal. For g-convexity conditions (1.1)–(1.3) are not equivalent (note that (1.3) implies (1.4) for any convexity in graphs [17]). Several characterizations for graphs with property (1.1), (1.2), or (1.3) are given in [14, 17, 27]. Here we will mention only one result which clearly shows an analogy with chordal graphs. Namely, a graph G fulfills the condition (1.3) iff G is a bridged graph, i.e., a graph which contains no isometric cycles of length at least four.

Note that a vertex is simplicial iff it is not midpoint of a P_3. Jamison and Olariu relaxed this condition in [19] in the following way: A vertex is *semisimplicial* iff it is not a midpoint of a P_4, and *nonsemisimplicial* otherwise. An ordering $(v_1,\dots,v_n)$ is a *semisimplicial ordering* iff v_i is semisimplicial in G_i for all $i=1,\dots,n$. In [19] the authors characterized the graphs for which every LexBFS-ordering is a semisimplicial ordering as the HHD-free graphs, i.e., the graphs which contain no house, hole, or domino as an induced subgraph (cf. Figure 1). Moreover, the graphs for which every MCS-ordering of an arbitrary induced subgraph F is a semisimplicial ordering of F are the HHP-free graphs, i.e., the graphs which contain no house, hole, or "P" as an induced subgraph (cf. Figure 1).

If a HHD-free graph G does not contain the "A" of Figure 1 as an induced subgraph then G is called *weak bipolarizable* (HHDA-free) [23].

In this paper we introduce the notion of m^3-convexity (a relaxation of m-convexity), which is closely related to semisimpliciality. A subset $S \subseteq V$ is called *m^3-convex* iff

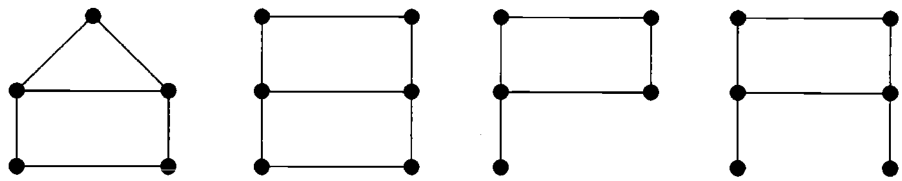


FIG. 1.

for any pair of vertices x, y of S each induced path of length at least 3 connecting x and y is completely contained in S. Note that a m^3-convex set is not necessarily connected, and it is not difficult to see that the family of m^3-convex sets is closed under intersection. Observe also that a vertex v is an extreme point of a m^3-convex set S iff v is semisimplicial in $G(S)$.

In this paper we present new characterizations of HHD-free and HHDA-free graphs via m^3-convexity. We show that for m^3-convexity the conditions (1.1)–(1.4) are again equivalent and hold iff the graph is HHD-free. We characterize weak bipolarizable graphs as the graphs for which the m^3-convex alignment is a convex geometry, i.e., by Theorem 1.1, for which every m^3-convex set is reachable via some semisimplicial ordering. Again, as for chordal graphs, in weak bipolarizable graphs every nonsemisimplicial vertex lies on an induced path of length at least 3 between semisimplicial vertices.

Convexity in graphs is a useful tool not only for geometric characterizations of several graph classes but also for resolving some problems related to distances in graphs [1, 4, 5, 6, 9, 14, 22]. As an application of our results we present a simple efficient criterion for deciding whether a HHD-free graph $G = (V, E)$ with given vertex radius function $r : V \to \mathbb{N}$ has an r-dominating clique. Note that this problem is $\mathbb{NP}$-complete for weakly chordal graphs (i.e., the graphs without holes and antiholes) [2]. From this criterion we obtain the inequality $diam(G) \geq 2rad(G) - 2$ between the diameter and radius of a HHD-free graph G. These results extend the known ones for chordal, distance-hereditary, and house-hole-domino-sun–free graphs [3, 5, 8, 9, 10].

Thus, the results of the paper show strict analogies between these graphs and chordal graphs. HHD-free, HHDA-free, and HHP-free graphs are three very natural generalizations of the class of chordal graphs.

2. m^3-convex sets in HHD-free graphs. In this section we characterize HHD-free graphs as the graphs with m^3-convex disks. Using m^3-convexity we give new properties of LexBFS-and MCS-orderings in HHD-free graphs and obtain known results from [19] as corollaries.

Since a vertex v is an extreme point of a m^3-convex set S iff v is semisimplicial in $G(S)$, we immediately conclude the following.

LEMMA 2.1. *An ordering $(v_1, \ldots, v_n)$ of the vertices of a graph G is semisimplicial iff $V(G_i) = \{v_i, \ldots, v_n\}$ is m^3-convex in G for all $i = 1, \ldots, n$.*

The following lemma will be frequently used in what follows.

LEMMA 2.2 (cycle lemma for hole-free graphs). *Let C be a cycle of length at least 5 in a hole-free graph G. Then for each edge xy of C there are vertices w_1, w_2 in C such that $xw_1 \in E$, $yw_2 \in E$, and $d(w_1, w_2) \leq 1$, i.e., each edge of a cycle is contained in a triangle or a 4-cycle.*

Proof. By induction on the length of the cycle. ∎

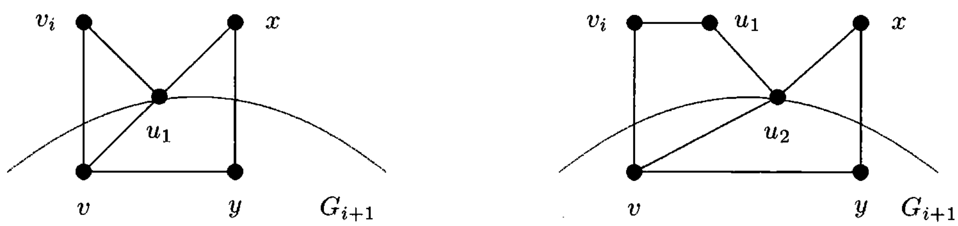


FIG. 2.

To make the paper self-contained we present the rules of the LexBFS and MCS algorithms.

LexBFS: Order vertices of a graph by assigning numbers from $n = |V|$ to 1. Assign the number k to a vertex v (as yet unnumbered), which has lexically largest vector $(s_i : i = n, n-1, \ldots, k+1)$, where $s_i = 1$ if v is adjacent to the vertex numbered i, and $s_i = 0$ otherwise.

MCS: Order vertices of a graph by assigning numbers from $n = |V|$ to 1. As the next vertex to number pick a vertex adjacent to the most numbered vertices.

Subsequently, we will write $x < y$ whenever in a given ordering of the vertex set of a graph G vertex x has a smaller number than vertex y.

In what follows we will often use the following properties:

$(P1)$ If $a < b < c$ and $ac \in E$ and $bc \notin E$, then there exists a vertex d such that $c < d$, $db \in E$, and $da \notin E$.

$(P2)$ If $a < b < c$ and $ac \in E$ and $bc \notin E$, then there exists a vertex d such that $b < d$, $db \in E$, and $da \notin E$.

Evidently, $(P2)$ is a relaxation of $(P1)$. It is well known that any LexBFS-ordering has property $(P1)$ [18] and any MCS-ordering has property $(P2)$ [28].

THEOREM 2.3.

(1) *Let G be a HHD-free graph and $(v_1, \ldots, v_n)$ be a LexBFS-ordering of G. Then for each $i = 1, \ldots, n$ the set $V(G_i)$ is m^3-convex in G.*

(2) *Let G be a HHP-free graph and $(v_1, \ldots, v_n)$ be a MCS-ordering of G. Then for each $i = 1, \ldots, n$ the set $V(G_i)$ is m^3-convex in G.*

Proof. We prove assertion (1) by induction on i. Assume that $V(G_i)$ is not m^3-convex in G but $V(G_j)$ is so for $j \geq i+1$. Then there must be a vertex y in G_{i+1} and an induced path P of length at least 3 connecting v_i and y, which contains some vertices not in G_i. Choose y and P such that $|P|$ is minimum and y is rightmost in the LexBFS-ordering.

Case 1. The neighbor of y in P does not belong to G_i.

Let x be this neighbor of y, and let $P = v_i - u_1 - \cdots - u_l - x - y$, $l \geq 1$. By applying $(P1)$ to $x < v_i < y$, we obtain a vertex $v > y$ adjacent to v_i but not to x.

The path $Q = v - v_i - u_1 - \cdots - u_l - x - y$ has both endpoints in G_{i+1}. By the induction hypothesis $V(G_{i+1})$ is m^3-convex. Thus Q cannot be induced. Since P is induced, all possible chords of Q must be incident to v. If v is adjacent only to y, we obtain a forbidden induced cycle of length at least 5. So let u_j be the vertex of $P \setminus \{y\}$ closest to y on the path P and adjacent to v. We immediately conclude $j = l$ for otherwise we have a hole. Now the m^3-convexity applied to $v - u_l - x - y$ implies $vy \in E$. Since the house and domino are forbidden subgraphs we conclude $l \geq 3$ (see Figure 2). Let $j < l$ be the index such that $vu_j \in E$, but $vu_s \notin E$ for all

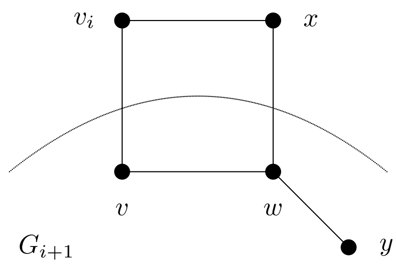


FIG. 3.

$s = j+1, \dots, l-1$. For $j = l-1$ we have a house; for $j = l-2$ we obtain a domino; otherwise $v - u_j - \cdots - u_l - v$ forms a hole.

Case 2. The neighbor of y in P belongs to G_i.

By minimality of $|P|$ we immediately conclude $P = v_i - x - w - y$, where $w, y \in V(G_{i+1})$ and $x \notin V(G_i)$. Now $(P1)$ applied to $x < v_i < w$ gives a vertex $v > w$ adjacent to v_i but not to x. We may choose v with maximum number in the LexBFS-ordering. By considering the path $v - v_i - x - w$ the m^3-convexity implies $vw \in E$. Note that $vy \notin E$ for otherwise we obtain a house. Therefore, we have constructed a "P" (see Figure 3).

Case 2.1. $y < v$.

By applying $(P1)$ to $v_i < y < v$ we obtain a vertex $u > v$ adjacent to y but not to v_i. Note that $w < v < u$ implies $u \neq w$. Suppose $ux \in E$. Then $(P1)$ applied to $x < v_i < u$ gives a vertex $t > u > v$ adjacent to v_i but not to x, a contradiction to the maximality of v. Thus $ux \notin E$. In the path $v - w - y - u$ both endpoints have greater numbers than y. Let $y = v_j$ for some $j > i$. Then the m^3-convexity of G_{j+1} implies $uv \in E$ or $uw \in E$. If we have both edges, then we obtain a house induced by $\{v_i, x, v, w, u\}$. If $uv \in E$ but $uw \notin E$ then we have a domino. Finally, if $uw \in E$ and $uv \notin E$ then we can replace y by $u > y$ in P, a contradiction to the choice of y.

Case 2.2. $y > v$.

By applying $(P1)$ to $w < v < y$ we obtain a vertex $u > y$ adjacent to v but not to w. If $uv_i \in E$ then m^3-convexity implies the edges ux and uy. So $\{v_i, u, x, y, w\}$ induces a house. Thus $uv_i \notin E$. Moreover, with the same arguments as in Case 2.1 we show $ux \notin E$. In the path $u - v - w - y$ both endpoints have greater numbers than v. Let $v = v_j$ for some $j > i$. Then the m^3-convexity of G_{j+1} implies $uy \in E$. Thus we get a domino. This settles the proof of assertion (1).

Now to get a proof for assertion (2) we can repeat the arguments of the proof above up to Cases 2.1 and 2.2 using $(P2)$ instead of $(P1)$. ∎

Note that any vertex $u \in V \setminus V(G_i)$ is semisimplicial in $G(\{u, v_i, \dots, v_n\})$ since $V(G_i)$ is m^3-convex in G. Thus we can conclude the following.

COROLLARY 2.4 (see [19]).

(1) *For any HHD-free graph G and any LexBFS-ordering $(v_1, \dots, v_n)$ of G vertex v_i is semisimplicial in G_i, $i = 1, \dots, n$.*

(2) *For any HHP-free graph G and any MCS-ordering $(v_1, \dots, v_n)$ of G vertex v_i is semisimplicial in G_i, $i = 1, \dots, n$.*

Moreover, since there is a MCS-ordering of the "P," which is not a semisimplicial ordering and neither holes nor a domino contain a semisimplicial vertex we immediately conclude the following.

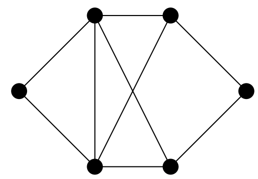


FIG. 4.

THEOREM 2.5 (see [19]). *A graph G is HHP-free iff any MCS-ordering of any induced subgraph F of G is a semisimplicial ordering of F.*

Note that in Theorem 2.5 it is necessary to consider all induced subgraphs of a given graph, since the graph presented in Figure 4 contains a "P" but every MCS-ordering of this graph is a semisimplicial ordering. For LexBFS it is sufficient to consider the graph itself, since as we will show the class of graphs where any LexBFS-ordering gives a semisimplicial ordering is a hereditary class.

A graph is called *nontrivial* if it has at least two vertices.

THEOREM 2.6. *The following conditions are equivalent for a graph G:*

(1) *G is HHD-free.*
(2) *Any LexBFS-ordering of G is a semisimplicial ordering.*
(3) *For any LexBFS-ordering $(v_1, \dots, v_n)$ of G the set $V(G_i)$ is m^3-convex in G for all $i = 1, \dots, n$.*
(4) *Every nontrivial induced subgraph of G has at least two semisimplicial vertices.*

Proof. It is easy to verify that none of a house, a domino, and holes contains two semisimplicial vertices. We have to show (2) $\Longrightarrow$ (1) and (2) $\Longrightarrow$ (4). All other directions are trivial or follow from Theorem 2.3.

(2) $\Longrightarrow$ (1) Let G be a graph such that every LexBFS-ordering is a semisimplicial ordering. Clearly, G cannot contain a hole or a domino since these graphs do not have a semisimplicial vertex. Assume that G contains a house induced by $\{a, b, c, d, e\}$ where $b - c - d - e - b$ induces a C_4 and a is adjacent to b and c. We start LexBFS at vertex a. By the rules of LexBFS both vertices d, e are smaller than the vertices b, c. Let v_i be the smaller one of d and e. Then v_i is not semisimplicial in G_i. Thus G is HHD-free.

(2) $\Longrightarrow$ (4) Let H be a nontrivial induced subgraph of G. Since H is HHD-free by (1) $\iff$ (2) there must be some semisimplicial vertex v of H. Now starting procedure LexBFS at v gives a second semisimplicial vertex. ∎

COROLLARY 2.7. *Let G be a HHD-free graph and v be a vertex of G. Then there is a semisimplicial vertex u such that $d(u, v) = e(v)$.*

Proof. We start procedure LexBFS at v. The first vertex u of the obtained LexBFS-ordering is semisimplicial by the above theorem and fulfills $d(u, v) = e(v)$ by the rules of LexBFS. ∎

We immediately conclude the following.

COROLLARY 2.8. *In any nontrivial HHD-free graph G there is a pair of semisimplicial vertices u, v such that $d(u, v) = diam(G)$.*

THEOREM 2.9. *The following conditions are equivalent for a graph G:*

(1) *G is HHD-free.*
(2) *The disk $D(v, 1)$ is m^3-convex for all vertices $v \in V$.*
(3) *The disks $D(v, k)$, $k \geq 1$, are m^3-convex for all vertices $v \in V$.*

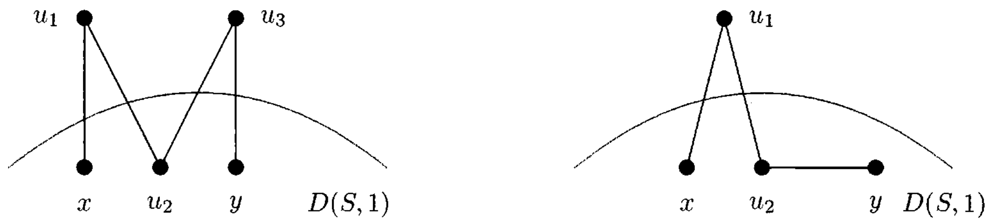


FIG. 5.

(4) *The set* $D(S,1) = \bigcup_{v \in S} D(v,1)$ *is* m^3*-convex for all connected sets* $S \subseteq V$.

(5) *The sets* $D(S,k) = \bigcup_{v \in S} D(v,k)$, $k \geq 1$, *are* m^3*-convex for all connected sets* $S \subseteq V$.

Proof. In every forbidden subgraph there is a vertex v such that $D(v,1)$ is not m^3-convex. So, we have to show only (1) $\Longrightarrow$ (5).

Suppose that there is a connected set S such that $D(S,1)$ is not m^3-convex. Then there are vertices x, y in $D(S,1)$ and there is an induced path $P = x - u_1 - \cdots - u_k - y$ such that $k \geq 2$ and at least one vertex u_i is not in $D(S,1)$. We may choose x, y, and P such that $|P|$ is minimal.

Case 1. $P \setminus \{x,y\} \subseteq V \setminus D(S,1)$.

We immediately conclude $x, y \notin S$. Moreover no u_i, $i = 1, \ldots, k$, is adjacent to some vertex of S. Let Q be a shortest path in $G(\{x,y\} \cup S)$ connecting x and y. Since $Q \setminus \{x,y\}$ is completely contained in S and both P and Q are induced, the cycle C formed by P and Q is chordless. From $|P| \geq 4$ we conclude $|C| \geq 5$—a contradiction.

Case 2. $|D(S,1) \cap P| \geq 3$.

By minimality of $|P|$, we obtain $k = 3$, $u_1, u_3 \notin D(S,1)$, and $u_2 \in D(S,1)$ or $k = 2$, $u_1 \notin D(S,1)$, and $u_2 \in D(S,1)$ (see Figure 5). Let $Q = x - z_1 - \cdots - z_l - y$, $l \geq 1$, be a shortest path in $G(\{x,y\} \cup S)$ connecting x and y and define $Q' := Q \setminus \{x,y\}$.

First consider the case $k = 2$. Note that $x, u_2 \notin S$, and u_1 is not adjacent to any vertex of Q'. Since the cycle $x - u_1 - u_2 - y - z_l - \cdots - z_1 - x$ is of length at least 5 the cycle lemma applied to the edge xu_1 gives $z_1u_2 \in E$. If $yz_1 \in E$ then we have a house. Hence $l \geq 2$. If $u_2z_2 \in E$ then we obtain a house. So let $u_2z_2 \notin E$. If y is adjacent to z_2 then we have a domino. Thus $l \geq 3$ and we can apply the cycle lemma to the edge z_1u_2 in the cycle $u_2 - y - z_l - \cdots - z_1 - u_2$ of length at least 5. So we conclude $u_2z_3 \in E$ which gives a domino.

Now consider the case $k = 3$. Note that $x, y, u_2 \notin S$. Since Q' is completely contained in S neither u_1 nor u_3 is adjacent to any vertex of Q'. On the other hand, the cycle $x - u_1 - u_2 - u_3 - y - z_l - \cdots - z_1 - x$ is of length at least 6. Thus the cycle lemma applied to the edge u_3y implies $u_2z_l \in E$. If $z_lx \notin E$ we proceed as in the case $k = 2$; otherwise we obtain a domino.

Thus, for every connected set S, $D(S,1)$ is m^3-convex. It is easy to see that $D(S,1)$ is connected too. Now, since $D(S,k) = D(D(S,k-1),1)$, we are done by induction on k. ◻

COROLLARY 2.10. *If in a HHD-free graph nonadjacent vertices* $x, y \in N^k(v)$ *are joined by a path* P *such that* $P \setminus \{x,y\}$ *is contained in* $V \setminus D(v,k)$, *then there is a common neighbor of* x *and* y *in* $N^{k+1}(v) \cap P$.

3. Weak bipolarizable graphs. Here we characterize weak bipolarizable graphs as the graphs for which the m^3-convex alignment is a convex geometry. Let $\mathfrak{M}^3(G)$

denote the set of all m^3-convex sets of a graph G. For a set $S \subseteq V$ the *m^3-convex hull m^3-conv(S)* is the smallest member of $\mathfrak{M}^3(G)$ containing S.

A set $H \subseteq V$ is *homogeneous* iff $N(x) \setminus H = N(y) \setminus H$ for any pair of vertices x, y of H. A homogeneous set H is *proper* iff $1 < |H| < |V|$.

The next lemma gives a nice criterion for checking the semisimpliciality of a vertex.

LEMMA 3.1. *A vertex v of a graph G is semisimplicial in G iff the connected components of the complement of $G(N(v))$ are homogeneous in G.*

Proof. If v is not semisimplicial then there is a P_4 containing v as midpoint, say $u_1 - v - u_2 - u_3$. Now u_1 and u_2 belong to a common connected component C of the complement of $G(N(v))$. But C is not homogeneous in G due to u_3.

To prove the converse let C be a connected component of the complement of $G(N(v))$ and suppose that C is not homogeneous in G. Then there must be vertices $x, y \in C$ and a vertex $z \in V \setminus C$ such that $xz \in E$ but $yz \notin E$. We may choose x and y such that their distance in the complement of $G(C)$ is minimal. Obviously, $z \neq v$. Moreover, since $yz \notin E$ but every vertex from $N(v) \setminus C$ must be adjacent to every vertex of C, we have $z \notin N(v)$. Thus $z \in N^2(v)$. If $xy \notin E$ then $z - x - v - y$ is a P_4. If $xy \in E$ then let $x - u_1 - \cdots - u_k - y$ be a shortest path in the complement of $G(C)$. Thus $xu_1 \notin E$. The minimal distance of x, y now implies $u_1 z \notin E$. Therefore, $z - x - v - u_1$ is a P_4. ∎

THEOREM 3.2 ([23]). *A graph G is weak bipolarizable iff each induced subgraph F of G is chordal or contains a proper homogeneous set.*

Let H be a proper homogeneous set in G and $v \in H$. Then the *homogeneous reduction $HRed(G, H, v)$* is the graph induced by $V(G) \setminus (H \setminus \{v\})$. Conversely, the *homogeneous extension $HExt(G, v, H)$* of G via a graph H in v with $V(H) \cap V(G) = \varnothing$ is the graph obtained by substituting v by H such that the vertices of H have the same neighbors outside of H as v had in G.

LEMMA 3.3. *Let H be a proper homogeneous set of a HHD-free graph G and $v \in H$.*

(1) *If x is semisimplicial in $HRed(G, H, v)$, but not in G, then $x \in H$, i.e., $x = v$.*
(2) *If $x \in H$ is semisimplicial in H, but not in G, then no vertex of H is semisimplicial in G and v is not semisimplicial in $HRed(G, H, v)$.*

Proof. Since no P_4 contains a proper homogeneous set, we conclude that for any 4-path P of G, either $P \subseteq H$ or $|P \cap H| \leq 1$.

(1) Since x is not semisimplicial in G it must be a midpoint of some 4-path P. If $x \notin H$ then the semisimplicity of x in $HRed(G, H, v)$ implies $|P \cap H| = 1$. But now we can replace the vertex of $P \cap H$ by v obtaining a P_4 in $HRed(G, H, v)$, which contains x as a midpoint—a contradiction. Thus $x \in H$, i.e., $x = v$.
(2) If $x \in H$ is semisimplicial in H, but not in G, then no P_4 in G containing x as a midpoint is completely contained in H. Thus $P \cap H = \{x\}$ for any 4-path P in G with midpoint x. Since H is homogeneous we can replace x in P by any vertex of H. Thus no vertex of H is semisimplicial in G, and v is not semisimplicial in $HRed(G, H, v)$. ∎

In [16] it is proved that in a chordal graph every nonsimplicial vertex lies on an induced path between two simplicial vertices. Next we present a stronger result which we will subsequently use.

LEMMA 3.4. *Let G be a chordal graph and $P = v_1 - \cdots - v_k$ be an induced path of length at least 2, i.e., $k \geq 3$. Then there are vertices u_i, $i = 1, \ldots, s$ and w_j, $j = 1, \ldots, t$, such that u_1, w_1 are simplicial and $u_1 - u_2 - \cdots - u_s - v_2 - \cdots - v_{k-1} -$*

$w_t - \cdots - w_2 - w_1$ *is an induced path in* G.

Proof. If both v_1 and v_k are simplicial then we are done. So suppose that v_1 is not simplicial.

Let M be the m-convex hull of $\{v_1, \ldots, v_k\}$ and S be the neighborhood of v_1 in M. Obviously, S is a $v_1 - v_3$-separator in M, i.e., v_1 and v_3 are in different connected components of $G(M) \setminus S$. We show that S is a $v_1 - v_3$-separator in G too. Assuming the contrary there must be an induced path P in $V \setminus S$ joining v_1 and v_3. Since S is the set of neighbors of v_1 in M the neighbor of v_1 in P does not belong to M. Thus P is an induced path between vertices of M which contains vertices of $V \setminus M$, a contradiction to the m-convexity of M. Therefore, S is a $v_1 - v_3$-separator in G.

Recall that every chordal graph is either complete or contains at least two nonadjacent simplicial vertices [7, 24]. Thus $G(M)$ as a chordal graph must contain at least two simplicial vertices. Since deleting a simplicial vertex from a m-convex set preserves m-convexity and since M is the m-convex hull of $\{v_1, \ldots, v_k\}$ we immediately conclude that v_1 and v_k are the only two simplicial vertices of M. Thus S is complete.

Since v_1 is not simplicial and all neighbors of v_1 are contained in $F := G(K \cup S)$, where K is the connected component of $G \setminus S$ containing v_1, the chordal graph F is not complete and hence there are two nonadjacent simplicial vertices in F. By the completeness of S at most 1 of them is in S. Thus we have a simplicial vertex u_1 in K which is simplicial in G too. Now consider a path P connecting the vertices v_1 and u_1 in K. Then no vertex up to v_2 of an induced subpath $u_1 - \cdots - u_s - v_2$ of the path $P \cup v_1v_2$ has a neighbor in $\{v_3, \cdots, v_k\}$. Hence, $u_1 - \cdots - u_s - v_2 - \cdots - v_k$ is an induced path. For v_k we proceed analogously. □

Note that every simplicial vertex is semisimplicial and thus, every nonsemisimplicial vertex is nonsimplicial.

LEMMA 3.5. *Every nonsemisimplicial vertex of a weak bipolarizable graph G lies on an induced path of length at least* 3 *between two semisimplicial vertices.*

Proof. We prove the assertion by induction on the size of G. The assertion holds for all graphs with at most 4 vertices since the only graph of these sizes which contains a nonsemisimplicial vertex is the P_4. Let x be a nonsemisimplicial vertex of G, i.e., x is a midpoint of some P_4.

If G is chordal then by Lemma 3.4 there is a path P of length at least 3 containing x such that both endpoints of P are simplicial and thus semisimplicial in G. Consequently, we are done.

Now assume that G is not chordal. Hence, by Theorem 3.2, G contains a proper homogeneous set H.

Case 1. $x \in H$.

Suppose that x is semisimplicial in $HRed(G, H, x)$. Then by Lemma 3.3 (2), vertex x is not semisimplicial in H. By the induction hypothesis x lies on an induced path of length at least 3 between semisimplicial vertices y, z in H. By Lemma 3.3 (2), both y and z must be semisimplicial in G too.

Now assume that x is not semisimplicial in $HRed(G, H, x)$. By the induction hypothesis x lies on an induced path between semisimplicial vertices y, z in $HRed(G, H, x)$. In particular, $y, z \notin H$. Thus by Lemma 3.3 (1), both y and z must be semisimplicial in G too.

Case 2. $x \notin H$.

From Lemma 3.3 (1) we immediately conclude that x is not semisimplicial in $HRed(G, H, v)$, where v is a semisimplicial vertex in the weak bipolarizable graph H.

By the induction hypothesis x lies on an induced path between semisimplicial vertices y, z in $HRed(G, H, v)$. Suppose that y is not semisimplicial in G. From Lemma 3.3 (1), we infer $y = v$. But now $y = v$ is not semisimplicial in $HRed(G, H, v)$ by Lemma 3.3 (2)—a contradiction. Thus both y and z are semisimplicial in G too. ∎

To prove the next corollary we use the arguments of the proof of [16, Corollary 3.4].

COROLLARY 3.6. *The Caratheodory number of the m^3-convex alignment of a weak bipolarizable graph is at most* 2.

Proof. Let $G = (V, E)$ be a weak bipolarizable graph and S be a subset of V. Pick an arbitrary vertex $x \in m^3\text{-}conv(S)$. If x is semisimplicial in the subgraph induced by $m^3\text{-}conv(S)$, then $x \in S$ since each extreme point of $m^3\text{-}conv(S)$ is in S by the definition of the hull of S. Otherwise, by Lemma 3.5, x lies on an induced path of length at least 3 between semisimplicial vertices of the subgraph induced by $m^3\text{-}conv(S)$. Hence, x is in the m^3-convex hull of two extreme points of $m^3\text{-}conv(S)$. Since each extreme point of $m^3\text{-}conv(S)$ is in S we are done. ∎

Subsequently, we call a vertex set S of G *reachable* iff there is an ordering $(v_1, \ldots, v_k)$ of $V \setminus S$ such that for each $i = 1, \ldots, k$ vertex v_i is semisimplicial in $G(\{v_i, \ldots, v_k\} \cup S)$.

THEOREM 3.7. *The following conditions are equivalent for a graph G:*

(1) *G is weak bipolarizable.*

(2) *In every induced subgraph F of G each nonsemisimplicial vertex lies on an induced path of length at least* 3 *between semisimplicial vertices of F.*

(3) *Each m^3-convex set of G is the hull of its semisimplicial vertices, i.e., $(V(G), \mathfrak{M}^3(G))$ is a convex geometry.*

(4) *A set S of G is m^3-convex iff there is an ordering $(v_1, \ldots, v_k)$ of $V(G) \setminus S$ such that for each $i = 1, \ldots, k$ vertex v_i is semisimplicial in $G(\{v_i, \ldots, v_k\} \cup S)$, i.e., S is reachable.*

Proof. We only need to prove (4) $\Longrightarrow$ (1).

Claim 1. If S is a m^3-convex set in $F := HRed(G, H, v)$, where H is a proper homogeneous set of G, then

$$S' := \begin{cases} S & : \quad v \notin S, \\ S \cup H & : \quad v \in S \end{cases}$$

is m^3-convex in G.

Suppose S' is not m^3-convex in G. Then there must be vertices $x, y \in S'$ and an induced path P of length at least 3 joining x and y such that $P \setminus S' \neq \varnothing$. If $|P \cap H| \leq 1$, then either P or $(P \setminus H) \cup \{v\}$ is an induced path in F of length at least 3 joining vertices of S which has at least one vertex outside S, a contradiction to the m^3-convexity of S in F. Now suppose $|H \cap P| \geq 2$. Note that $P \setminus H \neq \varnothing$. Let $P' = u_1 - \cdots - u_k$ be a maximal by inclusion subpath of P completely contained in H. Suppose $k \geq 2$. If $u_1 = x$ then $u_k \neq y$ since $P \setminus H \neq \varnothing$. Since H is homogeneous u_1 must be adjacent to the neighbor of u_k in $P \setminus P'$—a contradiction. If $u_1 \neq x$ then the same argument can be applied to u_k and the neighbor of u_1 in $P \setminus P'$. Now let $k = 1$. For $|H \cap P| \geq 2$ there must be a vertex $z \in H \cap P \setminus N(u_1)$. But now $N(u_1) \setminus H = N(z) \setminus H$ and $|P| \geq 4$ imply some chords in P, again a contradiction. Therefore, S' is m^3-convex in G.

Claim 2. Every homogeneous set H of a graph G is m^3-convex.

Let x, y be nonadjacent vertices of a homogeneous set H in G. If x has a neighbor z outside H then $yz \in E$, and vice versa. Thus any induced path between nonadjacent vertices of H containing vertices from $V \setminus H$ must be of length 2. Consequently, H is m^3-convex in G.

Claim 3. Let H be a proper homogeneous set of a graph G. If S is m^3-convex in $G(H)$ then it is so in G.

Since S is a subset of H we can use the same arguments as in the proof of Claim 2.

Claim 4. If v is a simplicial vertex in a graph G then any m^3-convex set of $G \setminus \{v\}$ is m^3-convex in G.

Since the neighborhood of a simplicial vertex v is complete no induced path of length at least 3 can contain v as an inner point.

Now we prove by induction on the size of G that any graph fulfilling (4) is weak bipolarizable, i.e., HHDA-free. Since any singleton of $V(G)$ is a m^3-convex set, G possesses a semisimplicial ordering, and thus does not contain a hole or a domino. Let F be an induced subgraph of G isomorphic to the house and K be the 3-clique of F. Now the m^3-convex set K must be reachable, but no vertex of $F \setminus K$ is semisimplicial in F—a contradiction. Therefore, G is a HHD-free graph.

Case 1. G contains a proper homogeneous set H.

Let v be a vertex of H, $F := HRed(G, H, v)$ and S be a m^3-convex set in F. Then S' as defined in Claim 1 is m^3-convex in G and thus reachable. Hence, S is reachable in F since each semisimplicial vertex of G is semisimplicial in every induced subgraph containing this vertex. Therefore, F fulfills (4) and, by the induction hypothesis, is HHDA-free. Applying the same arguments to a m^3-convex set S of H and using Claim 3 implies that H is HHDA-free. Now we conclude that G itself is HHDA-free as the homogeneous extension of the HHDA-free graph F by the HHDA-free graph H (see [23]).

Case 2. G has no proper homogeneous set.

Suppose G contains an "A" induced by the 4-cycle $x - c - d - y - x$ and the pendant vertices a, b where $ax \in E$ and $by \in E$. In what follows we prove that $M := D(a, 1) \cup D(b, 1)$ is m^3-convex in G. Thus M must be reachable, but neither c nor d are semisimplicial in the "A"—a contradiction.

First note that every semisimplicial vertex v of G is simplicial due to Lemma 3.1. From Claim 4 we conclude that $G \setminus \{v\}$ fulfills (4) and thus, by the induction hypothesis, is HHDA-free. Therefore, a and b are the only semisimplicial vertices of G, and $D(a, 1)$, $D(b, 1)$ are complete.

- *If there is a common neighbor z of a and b, then z is adjacent to all vertices a, b, c, d, x, y.*
  Considering the cycle $z - a - x - y - b - z$ implies the edges zx and zy. Now $\{z, x, y, c, d\}$ induces a house, thus $zc \in E$ or $zd \in E$. Suppose $zc \notin E$. Then $zd \in E$ and $\{a, z, x, c, d\}$ induces a house. Hence both $zc \in E$ and $zd \in E$.
- *$N(a) \subseteq N(c)$ and $N(b) \subseteq N(d)$.*
  Let w be a neighbor of a and suppose $wc \notin E$. Thus $w \neq x$, $wx \in E$, and $wb \notin E$. Since $G \setminus \{a\}$ is HHDA-free w must be adjacent to y or d. If $wy \in E$ then the graph induced by $\{w, x, y, c, d\}$ implies $wd \in E$. Hence $wd \in E$. But now $\{a, x, w, c, d\}$ induces a house.
- *Every vertex of $N(a)$ is adjacent to every vertex of $N(b)$.*
  If $w \in N(a) \cap N(b)$, then w is adjacent to all vertices of $N(a) \cup N(b)$ since both $D(a, 1)$ and $D(b, 1)$ are complete. So suppose for the contrary that there are nonadjacent vertices $z \in N(a) \setminus N(b)$ and $w \in N(b) \setminus N(a)$. Since $xy \in E$ we have either $z = x$ and $w \neq y$, $z \neq x$, and $w = y$ or $z \neq x$ and $w \neq y$.

First assume $z = x$ (analogously, $w = y$). The graph induced by $\{w, d, y, c, z\}$ implies $wc \in E$. But now $\{b, y, w, z, c\}$ induces a house. So let $x \neq z$ and $y \neq w$. By the same arguments as above we may assume $zy \in E$ and $wx \in E$. Now considering $\{w, d, y, z, c\}$ gives $zd \in E$ or $wc \in E$. By symmetry, say $wc \in E$. But this yields a house induced by $\{b, y, w, z, c\}$.

To complete the proof suppose that $M = D(a, 1) \cup D(b, 1)$ is not m^3-convex in G. Then there must be nonadjacent vertices $w, z \in M$ and an induced path P of length at least 3 joining w and z such that $P \setminus M$ is nonempty. Since every vertex of $N(a)$ is adjacent to every vertex of $N(b)$ we conclude $\{w, z\} \cap \{a, b\} \neq \emptyset$. Say $z = a$. Then $w \notin D(a, 1)$. Let z' be the neighbor of z in P, i.e., $z' \in N(a)$. If $w \in N(b)$ then $z'w \in E$ gives a contradiction. Hence $w = b$. Now consider the neighbor w' of w in P. From $w' \in N(b)$ we conclude $z'w' \in E$—again a contradiction. □

4. The existence of r-dominating cliques. Let $r : V \to \mathbb{N}$ be some vertex function defined on G. Then a set $D \subseteq V$ *r-dominates* G iff for all vertices x in $V \setminus D$ there is a vertex $y \in D$ such that $d(x, y) \leq r(x)$. D is a *r-dominating clique* iff D is complete and r-dominates G. Note that there are graphs and vertex functions r such that G has no r-dominating clique. For some graph classes, such as chordal, distance-hereditary, and HHDS-free graphs, there is an existence criterion for r-dominating cliques [9, 8, 10]. In what follows we prove this criterion for HHD-free graphs. The method is similar to the one used for chordal graphs in [9] and essentially exploits m^3-convexity of disks in HHD-free graphs.

LEMMA 4.1. *Let C be a clique in a HHD-free graph G and v be a vertex of G such that for all vertices w of C the distance to v is $k \geq 1$. Then there is a vertex u at distance $k - 1$ to v which is adjacent to all vertices of C.*

Proof. We prove the assertion by induction on k. For $k = 1$ there is nothing to show. Let x be a vertex of $N^{k-1}(v)$ adjacent to a maximal number of vertices of C. Suppose that there is some vertex $a \in C$ which is not adjacent to x, and let y be a neighbor of a in $N^{k-1}(v)$. By the choice of x there must be a vertex $b \in C$ adjacent to x but not to y. Thus we have the path $x - b - a - y$ of length 3 between vertices x, y of $D(v, k-1)$, which contains vertices a, b outside of $D(v, k-1)$. By Theorem 2.9 $D(v, k-1)$ is m^3-convex; hence $xy \in E$. Now, by applying the induction hypothesis to the clique $\{x, y\}$ we obtain a common neighbor u of x, y in $N^{k-2}(v)$. Therefore we have constructed a house—a contradiction. □

In a similar way we can prove the following lemma.

LEMMA 4.2. *If x, y, v are vertices of a HHD-free graph G such that $d(x, v) = d(y, v) = k$ and $N(x) \cap N(y) \cap N^{k+1}(v) \neq \emptyset$, then there is a vertex $u \in N(x) \cap N(y) \cap N^{k-1}(v)$.*

Define the *projection* of a vertex v to a set S by

$$Proj(v, S) := \{x \in S : d(v, x) = d(v, S)\}$$

and the projection of a set C to a set S by $Proj(C, S) := \bigcup_{v \in C} Proj(v, S)$.

LEMMA 4.3. *Let u, v be vertices of a HHD-free graph. Then for any vertex x in $D(v, k)$ there is a shortest path between u and x going through the projection $Proj(u, D(v, k))$.*

Proof. If $d(u, v) \leq k$ then $Proj(u, D(v, k)) = \{u\}$ and there is nothing to show. So let $d(u, v) \geq k + 1$. Choose an arbitrary vertex $w \in Proj(u, D(v, k))$ and assume $d(u, x) < d(u, w) + d(w, x)$. Let P be a shortest path connecting u and x, and let z be the vertex of $V(P) \cap D(v, k)$ closest to u on the path P (see Figure 6). Thus

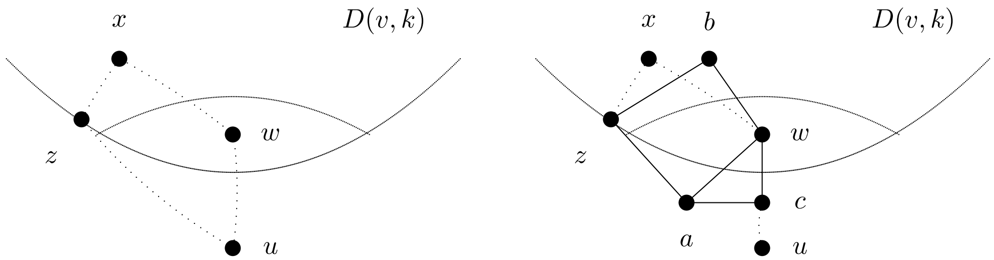


Fig. 6.

$d(u,x) = d(u,z) + d(z,x)$. If $z \in Proj(u, D(v,k))$ then we are done. So assume $z \notin Proj(u, D(v,k))$ implying $d(u,z) > d(u,w)$. Note that $zw \notin E$ for, otherwise,

$$d(u,w)+1+d(z,x) \leq d(u,z)+d(z,x) = d(u,x) < d(u,w)+d(w,x) \leq d(u,w)+d(z,x)+1$$

is a contradiction. Thus by Corollary 2.10 there is a common neighbor a of w and z in $N^{k+1}(v) \cap P$ implying that $d(u,z) \leq d(u,w) + 2$ and $d(w,x) \leq d(z,x) + 2$. Moreover, $d(u,x) < d(u,w) + 2 + d(z,x)$. Therefore, $d(u,z) + d(z,x) = d(u,x) < d(u,w) + 2 + d(z,x)$ gives $d(u,z) = d(u,w) + 1$, and $d(u,a) = d(u,w)$. Now applying Lemma 4.2 to z, w, and v gives a common neighbor b of z, w in $N^{k-1}(v)$. By distance requirements $ab \notin E$. Furthermore, Lemma 4.1, applied to $\{a, w\}$ and u, yields a common neighbor c of a, w at distance $d(u,w) - 1$ to u. Thus neither $cz \in E$ nor $cb \in E$. Consequently, $\{a, b, c, w, z\}$ induces a house. $\square$

Let U_1, U_2 be subsets of V. The sets U_1, U_2 form a *join* iff any vertex of U_1 is adjacent to any vertex of U_2.

Lemma 4.4. *Let G be a HHD-free graph and xy be an edge outside of $D(v,k)$. Then $Proj(x, D(v,k)) \subseteq Proj(y, D(v,k))$ or $Proj(y, D(v,k)) \subseteq Proj(x, D(v,k))$. Moreover, assuming $Proj(x, D(v,k)) \subseteq Proj(y, D(v,k))$ implies that the sets $Proj(x, D(v,k))$ and $Proj(y, D(v,k)) \setminus Proj(x, D(v,k))$ form a join.*

Proof. We will present the proof for the equidistant case, i.e., $d(x,v) = d(y,v)$. The cases $d(x,v) = d(y,v)+1$ and $d(y,v) = d(x,v)+1$ can be handled in a similar (even easier) way. Let $A := Proj(x, D(v,k)) \cap Proj(y, D(v,k))$, $B := Proj(x, D(v,k)) \setminus A$, and $C := Proj(y, D(v,k)) \setminus A$.

Suppose $w_x \in B$, $w_y \in C$. Since $d(y, w_y) = d(x, w_x) = d(x,v) - k$ we have $d(x, w_y) = d(x, w_x) + 1$ and $d(y, w_x) = d(y, w_y) + 1$. Now if $w_x w_y \notin E$ we get a contradiction to Corollary 2.10. Therefore, $w_x w_y \in E$. Let b (c) be the neighbor of w_x (w_y) in a shortest path P_x (P_y) between x (y) and w_x (w_y). Obviously, $w_x c, w_y b \notin E$. Lemma 2.2 applied to the edge $w_x w_y$ in the cycle induced by the vertices of P_x and P_y gives $bc \in E$. Thus $\{b, c, w_x, w_y, s\}$ induces a house where s is a common neighbor of $w_x w_y$ in $N^{k-1}(v)$ due to Lemma 4.1. Consequently, either $B = \varnothing$ or $C = \varnothing$.

Finally, suppose $w \in A$, $w_x \in B$ and $w_x w \notin E$. Consider the three vertices w, w_x, v. By Corollary 2.10 there is a common neighbor z of w and w_x at distance $k+1$ to v and $d(x,w) - 1$ to x. By Lemma 4.2 there is a common neighbor u of w and w_x at distance $k-1$ to v. Let t be the neighbor of w on a shortest path joining w and y. Since $w_x \notin A$ we have $tw_x \notin E$. By distance requirements $zu, tu \notin E$. If $tz \in E$ then $\{t, z, w, w_x, u\}$ induces a house. So assume $tz \notin E$ and consider the cycle C formed by w and by the shortest paths joining t, y and z, x. Obviously $|C| \geq 5$. Applying the circle lemma to edge zw yields the edge ts, where s is the neighbor of

z in the shortest path between x and z. By distance requirements $\{s,t,z,w,w_x,u\}$ induces a domino. Therefore, A and B form a join. □

LEMMA 4.5. *Let G be a HHD-free graph and C be a clique such that $C \setminus D(v,k) \neq \varnothing$. Then there is some vertex $u \in N^{k-1}(v)$ adjacent to all vertices of $Proj(C, D(v,k))$.*

Proof. Choose a maximal clique C' in $Proj(C, D(v,k))$ containing $C \cap D(v,k)$. By Lemma 4.1 there is a vertex a in $N^{k-1}(v)$ adjacent to all vertices of C'. Choose such a vertex a with a maximal number of neighbors in $Proj(C, D(v,k))$ and suppose that there is some vertex $y \in Proj(C, D(v,k)) \setminus C'$ nonadjacent to a. Since C' is maximal there is a vertex $w \in C'$ which is not adjacent to y. Note $y \notin C$. Thus there is a common neighbor z of y and w in $N^{k+1}(v)$ (either $z \in C$ or the existence of z follows from Corollary 2.10). Now applying Lemma 4.2 to w, y gives a common neighbor b of w and y in $N^{k-1}(v)$. By distance requirements $za, zb \notin E$. If $ab \in E$, then $\{a,b,y,z,w\}$ induces a house. If $ab \notin E$, then we can apply Lemma 4.2 to a, b yielding a common neighbor c of a, b in $N^{k-2}(v)$. But now $\{c,a,b,y,w,z\}$ induces a domino. □

THEOREM 4.6. *Let G be a HHD-free graph and $r : V \to \mathbb{N}$ be a vertex function on G. Then G has a r-dominating clique iff for all vertices $x, y \in V$, $d(x,y) \leq r(x) + r(y) + 1$ holds.*

Proof. Obviously, if G has a r-dominating clique then the inequality is fulfilled. To prove the converse let $(v_1, \dots, v_n)$ be any ordering of V and suppose that there is a clique C which r-dominates $\{v_1, \dots, v_{i-1}\}$ but not v_i. Thus $d(v_i, C) \geq r(v_i) + 1$. Let $B := Proj(C, D(v_i, r(v_i) + 1))$.

Claim 1. B r-dominates $\{v_1, \dots, v_{i-1}\}$.

Let $k \leq i-1$ and consider vertex v_k. Since C r-dominates $\{v_1, \dots, v_{i-1}\}$ there is some vertex $c \in C$ such that $d(c, v_k) \leq r(v_k)$.

If $v_k \in D(v_i, r(v_i) + 1)$ then by Lemma 4.3 there is a shortest path joining c and v_k going through B. Thus v_k is r-dominated by some vertex of B.

Now let $v_k \in V \setminus D(v_i, r(v_i)+1)$. Since $d(v_k, v_i) \leq r(v_k) + r(v_i) + 1$ we may choose a vertex x_k in $D(v_k, r(v_k)) \cap N^{r(v_i)+1}(v_i)$. Again, by Lemma 4.3 there is a shortest path joining c and x_k which contains a vertex of B, say y_k. If $d(c, x_k) \geq 3$, then $y_k \in D(v_k, r(v_k))$ since both c and x_k are contained in the m^3-convex set $D(v_k, r(v_k))$. If $cx_k \in E$ then either $c = y_k$ or $x_k = y_k$ and we are done since v_k is r-dominated by y_k. So let $d(c, x_k) = 2$. Again, if $c = y_k$ or $x_k = y_k$ then we are done. Thus let $c - y_k - x_k$ induce a P_3 and assume $d(v_k, y_k) > r(v_k)$. We immediately conclude $d(v_k, c) = d(v_k, x_k) = r(v_k)$ and $d(v_k, y_k) = r(v_k) + 1$. Thus Lemma 4.2 applied to c, x_k, and v_k gives a common neighbor a of c and x_k at distance $r(v_k) - 1$ to v_k. Since

$$d(v_i, y_k) = d(v_i, x_k) = d(v_i, a) - 1 = d(v_i, c) - 1 = r(v_i) + 1$$

applying Lemma 4.1 to the edge $x_k y_k$ and to v_i yields a common neighbor b of x_k and y_k at distance $r(v_i)$ to v_i. By distance requirements the set $\{a, b, x_k, y_k, c\}$ induces a house—a contradiction. Thus y_k r-dominates v_k and we are done.

Let C'' be a maximal clique in $Proj(C, D(v_i, r(v_i) + 1))$ such that $C'' \supset C \cap D(v_i, r(v_i) + 1)$. By Lemma 4.5 there is a vertex a in $N^{r(v_i)}(v_i)$ adjacent to all vertices of B. Define $C' := C'' \cup \{a\}$.

Claim 2. C' r-dominates $\{v_1, \dots, v_i\}$.

Obviously, a r-dominates v_i. Suppose there is some vertex v_k, $k \leq i-1$ which is not r-dominated by C'. By Claim 1 v_k is r-dominated by B. More exactly, there is a vertex $c \in C$ and a vertex $y_k \in Proj(c, D(v_i, r(v_i) + 1)) \subseteq B \setminus C'$ both r-dominating

v_k. Since C'' is maximal there must be a vertex $w \in C''$ nonadjacent to y_k. By Lemma 4.4 both vertices y_k, w are contained in the projection of c.

Let z be a common neighbor of w and y_k at distance $d(c,w)-1$ to c obtained from Lemma 4.2. If $d(y_k,c) \geq 3$ then the m^3-convexity of $D(v_k, r(v_k))$ implies $z \in D(v_k, r(v_k))$. We conclude $d(v_k,z) = d(v_k,y_k) = r(v_k)$. Now we can apply Lemma 4.1 to the edge $y_k z$ obtaining a common neighbor s of y_k and z at distance $r(v_k)-1$ to v_k. By distance requirements $sw, sa, az \notin E$. Thus $\{s,w,a,z,y_k\}$ induces a house. In a similar way we can handle the case $c = z$. So assume $d(y_k,c) = 2$. If $z \in D(v_k, r(v_k))$ then we proceed as above. So by assuming $d(z,v_k) > r(v_k)$ we have $d(v_k,c) = d(v_k,y_k) = r(v_k)$ and $d(v_k,z) = r(v_k)+1$. Now we can apply Lemma 4.2 to c, y_k obtaining a common neighbor b of c, y_k at distance $r(v_k)-1$ to v_k. By distance requirements $bw, ba \notin E$. Thus $\{c,b,z,y_k,a,w\}$ induces a domino.

Consequently we have constructed a clique which r-dominates $\{v_1,\ldots,v_i\}$. Induction on i settles the proof. ∎

COROLLARY 4.7. *For a HHD-free graph G we have $2rad(G) \geq diam(G) \geq 2(rad(G)-1)$.*

Proof. Suppose that $diam(G) < 2(rad(G)-1)$. Then by Theorem 4.6 for $r(v) := rad(G)-2$, $v \in V$, there exists a r-dominating clique C in G. Hence, any vertex v of C has $e(v) \leq rad(G)-1$, a contradiction to the definition of the radius. ∎

Acknowledgments. The authors are indebted to two anonymous referees for constructive comments improving the presentation.

REFERENCES

[1] A. Brandstädt, V.D. Chepoi, and F.F. Dragan, *Perfect elimination orderings of chordal powers of graphs*, Discrete Math., 158 (1996), pp. 273–278.
[2] A. Brandstädt and D. Kratsch, *Domination problems on permutation and other graphs*, Theoret. Comput. Sci., 54 (1987), pp. 181–198.
[3] G.J. Chang and G.L. Nemhauser, *The k-domination and k-stability problems on sun-free chordal graphs*, SIAM J. Algebraic Discrete Meth., 5 (1984), pp. 332–345.
[4] V.D. Chepoi, *d-Convex Sets in Graphs*, Ph.D. dissertation, Moldova State University, Kishinev, 1986 (Russian).
[5] V.D. Chepoi, *Centers of triangulated graphs*, Math. Notes, 43 (1988), pp. 143–151 (Russian, English transl.).
[6] V.D. Chepoi and F.F. Dragan, *Linear-time algorithm for finding a central vertex of a chordal graph*, Algorithms – ESA'94, 2nd Annual European Symposium, Utrecht, The Netherlands, 1994 Lecture Notes in Comput. Sci. 855, Jan van Leeuwen, ed., Springer, Berlin, New York, 1994, pp. 159–170.
[7] G.A. Dirac, *On rigid circuit graphs*, Abh. Math. Sem. Univ. Hamburg, 25 (1961), pp. 71–76.
[8] F.F. Dragan, *Dominating cliques in distance-hereditary graphs*, Algorithm Theory – SWAT'94, 4th Scandinavian Workshop on Algorithm Theory, Aarhus, Denmark, 1994, Lecture Notes in Comput. Sci. 824, Erik M. Schmidt and Sven Skyum, eds., Springer, Berlin, New York, 1994, pp. 370–381.
[9] F.F. Dragan and A. Brandstädt, *r-dominating cliques in graphs with hypertree structure*, Discrete Math., 162 (1996), pp. 93–108.
[10] F.F. Dragan and F. Nicolai, *r-domination problems on homogeneously orderable graphs*, Networks, 30 (1997), pp. 121–131.
[11] F.F. Dragan, F. Nicolai, and A. Brandstädt, *Convexity and HHD-free graphs*, Abstracts of the 5th Internat. Colloq. on "Graphs and Combinatorics," Marseille Luminy, September 3–8, 1995, p. 17.
[12] P. Duchet, *Convex sets in graphs* II: *Minimal path convexity*, J. Combin. Theory Ser. B, 44 (1988), pp. 307–316.
[13] P. Duchet and H. Meyniel, *Ensemble convexes dans les graphes* I, European J. Combin., 4 (1983), pp. 127–132.
[14] M. Farber, *Bridged graphs and geodesic convexity*, Discrete Math., 66 (1987), pp. 249–257.

[15] M. Farber, *On diameters and radii of bridged graphs*, Discrete Math., 73 (1989), pp. 249–260.
[16] M. Farber and R.E. Jamison, *Convexity in graphs and hypergraphs*, SIAM J. Algebraic Discrete Meth., 7 (1986), pp. 433–444.
[17] M. Farber and R.E. Jamison, *On local convexity in graphs*, Discrete Math., 66 (1987), pp. 231–247.
[18] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.
[19] B. Jamison and S. Olariu, *On the semi-perfect elimination*, Adv. Appl. Math., 9 (1988), pp. 364–376.
[20] R.E. Jamison, *A perspective on abstract convexity: Classifying alignment by varieties*, in Convexity and Related Combinatorial Geometry, D.C. Kay and M. Breen, eds., Marcel Dekker, New York, 1982.
[21] B. Korte, L. Lovász, and R. Schrader, *Greedoids*, Springer-Verlag, Berlin, Heidelberg, New York, 1991.
[22] H.M. Mulder, *The Interval Function of a Graph*, preprint, Math. Centre Tracts 132, Math. Centrum, Amsterdam, 1980.
[23] S. Olariu, *Weak bipolarizable graphs*, Discrete Math., 74 (1989), pp. 159–171.
[24] D. Rose, *Triangulated graphs and the elimination process*, J. Math. Anal. Appl., 32 (1970), pp. 579–609.
[25] D. Rose, R.E. Tarjan, and G. Lueker, *Algorithmic aspects on vertex elimination on graphs*, SIAM J. Comput., 5 (1976), pp. 266–283.
[26] V.P. Soltan, *d-convexity in graphs*, Soviet Math. Dokl., 28 (1983), pp. 419–421.
[27] V.P. Soltan and V.D. Chepoi, *Conditions for invariance of set diameters under d-convexification in a graph*, Cybernetics, 19 (1983), pp. 750–756.
[28] R.E. Tarjan and M. Yannakakis, *Simple linear time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs*, SIAM J. Comput., 3 (1984), pp. 566–579.

SIAM J. DISCRETE MATH.
Vol. 12, No. 1, pp. 136–153

CONSTRAINING PLANE CONFIGURATIONS IN COMPUTER-AIDED DESIGN: COMBINATORICS OF DIRECTIONS AND LENGTHS *

BRIGITTE SERVATIUS† AND WALTER WHITELEY‡

Abstract. Configurations of points in the plane constrained by directions only or by lengths alone lead to equivalent theories known as parallel drawings and infinitesimal rigidity of plane frameworks. We combine these two theories by introducing a new matroid on the edge set of the complete graph with doubled edges to describe the combinatorial properties of direction-length designs.

Key words. computer-aided design, constraint frameworks, generic rigidity, matroid, parallel drawings, plane configurations

AMS subject classifications. Primary, 68U07, 05B35; Secondary, 05C50, 51N05, 52C25

PII. S0895480196307342

1. Introduction.

A *plane configuration* in computer-aided design (CAD) is a collection of geometric objects such as points, line segments, and circular arcs in the plane, together with constraints on and between these objects [7, 13]. Naturally the designer wants to know if a realization of the configuration exists and is uniquely determined. A realization of a plane configuration is called a *plane design*. Beyond simple uniqueness of design, there are other fundamental design questions: If global uniqueness is not achieved, is the design locally unique? If the design permits continuous deformations, which additional constraints would give the appropriate uniqueness? Are all constraints essential in producing the design or are there constraints which are forced by the remaining ones?

Given a design, the constraints can be written as a system of algebraic equations whose variables are the coordinates and parameters of the geometric objects [12, 15]. Some of the above questions may be answered by computing the rank of the Jacobian of the system of constraint equations [13, 15]. Because of the size of the system and possible degeneracies, computation may be slow and unstable. Therefore a mathematical theory which answers these questions purely combinatorially is desirable [3, 12, 18].

The classical problem of Euclidean construction may be stated in the language of plane designs, as well as other familiar geometric problems. Much is known about *length designs*, where the objects are points and the distances between certain pairs of points are prescribed, forming the familiar mathematical model for a bar and joint framework [8]. On the other hand, *direction designs*, in which the constraints prescribe directions instead of distances between points, are also well understood as the problem of parallel drawings [17]. We present a combinatorial solution for the Jacobian of direction-length designs, which incorporate both of these cases.

*Received by the editors July 26, 1996; accepted for publication (in revised form) June 9, 1998; published electronically January 29, 1999.

http://www.siam.org/journals/sidma/12-1/30734.html

†Department of Mathematics, Worcester Polytechnic Institute, Worcester, MA 01609-2280 (bservat@wpi.edu).

‡Department of Mathematics and Statistics, York University, 4700 Keele Street, North York, ON, M3J 1P3, Canada (whiteley@mathstat.yorku.ca). The research of this author was supported by grants from NSERC (Canada).

These results are a contribution to the more basic open case of lengths and angles, a problem which arises in geodesy (making maps).

We will start out by summarizing results for frameworks and parallel drawings in section 2, then define direction-length designs in section 3. Our main goal is to characterize robust designs (defined in section 4), which have independent constraints and locally unique realizations. Limiting designs are used as tools in our proofs and are explicitly described in section 5. In section 6 we describe a direction-length construction and prove that the construction produces robust designs. The converse is demonstrated in section 6, where the combinatorial properties of direction-length designs are explored. Finally we indicate problems arising from mixing lengths, directions, and angles and outline other topics for further research.

2. Frameworks and parallel drawings.

2.1. Frameworks. Consider the set $V = \{1, \ldots, n\}$ and a function $\mathbf{p}$ from V into $\mathbb{R}^2$. We call $\mathbf{p}$ a *configuration* and we will denote $\mathbf{p}(i)$ by $\mathbf{p}_i$. A configuration $\mathbf{p}$ is *generic* if the coordinates in $\mathbf{p}$ are algebraically independent over the rationals. (For convenience here, we will assume that all points in a configuration are distinct, $\mathbf{p}_i \neq \mathbf{p}_j, i \neq j$. In certain limiting cases, we will bring vertices into coincidence *and* redefine the associated constraint.)

If $\mathbf{p}$ is an embedding, we can associate with every graph $G = (V, E)$ a *framework* $G(\mathbf{p})$, where the edge set E is interpreted as the collection of those pairs of vertices whose images under $\mathbf{p}$ are joined by rigid bars. We call two frameworks $G(\mathbf{p})$ and $G(\mathbf{q})$ equivalent if corresponding bars have the same length.

We may identify the configuration $\mathbf{p}$ with a point in $\mathbb{R}^{2n}$, and measure the distance between pairs of vertices by evaluating the *rigidity function* $\rho : \mathbb{R}^{2n} \to \mathbb{R}^{n(n+1)/2}$ defined by $\rho(\mathbf{p})_{i,j} = (\mathbf{p}_i - \mathbf{p}_j)^2$ for $i < j \leq |V|$. Clearly ρ is continuously differentiable with respect to $\mathbf{p}$, and we define $R(\mathbf{p})$, the rigidity matrix for the configuration $\mathbf{p}$, by $\rho'(\mathbf{p}) = 2R(\mathbf{p})$. With every framework $G(\mathbf{p})$ we can associate the matrix $R(G, \mathbf{p})$ consisting of those rows of $R(\mathbf{p})$ corresponding to E. A solution, $\mathbf{u}$, of the system $R(G, \mathbf{p})\mathbf{u} = \mathbf{0}$ consists of vectors $\mathbf{u}_i$ in $\mathbb{R}^2$, one for each point $\mathbf{p}_i$ satisfying

$$(\mathbf{p}_i - \mathbf{p}_j) \cdot (\mathbf{u}_i - \mathbf{u}_j) = 0$$

for each $(i, j) \in E$. $\mathbf{u}$ is called an *infinitesimal motion* of the framework. If $|V| \geq 2$ and $R(G, \mathbf{p})$ has rank $2n - 3$, or equivalently if all solutions to $R(G, \mathbf{p})\mathbf{x} = \mathbf{0}$ correspond to derivatives of congruences (translations or rotations), the framework is called *infinitesimally rigid*. An infinitesimally rigid framework with independent rows of the rigidity matrix is called *isostatic*.

A configuration $\mathbf{p}$ is said to be *generic* if any length design whose constraints are dependent with respect to $\mathbf{p}$ are in fact dependent with respect to any embedding. It is straightforward to show that almost all embeddings are generic (see [2]). If the coordinates of $\mathbf{p}$ are algebraically independent over the rational field, then $\mathbf{p}$ is generic. For a generic embedding, the linear independence of the rows of the rigidity matrix depends only on the graph whose edges correspond to the rows, and consequently, the *generic rigidity* of a framework depends on the graph alone.

2.2. Parallel drawings. If $\mathbf{u}$ is an infinitesimal motion of $R(G)$, $\mathbf{u}_i = (u_i, v_i)$, then $\mathbf{u}_i^\perp - \mathbf{u}_j^\perp$ is parallel to $\mathbf{p}_i - \mathbf{p}_j$ for every edge (i, j), where $\mathbf{u}_i^\perp = (v_i, -u_i)$; so $G(\mathbf{p}+\mathbf{u}^\perp)$ is a framework whose edges are all parallel to edges in $G(\mathbf{p})$ (see Figure 2.1). $G(\mathbf{p}+\mathbf{u}^\perp)$ is said to be a *parallel redrawing* of $G(\mathbf{p})$. If $\mathbf{t}$ is an infinitesimal translation, then $G(\mathbf{p}+\mathbf{t}^\perp)$ is congruent to $G(\mathbf{p})$. If $\mathbf{r}$ is an infinitesimal rotation, $G(\mathbf{p}+\mathbf{r}^\perp)$ is a

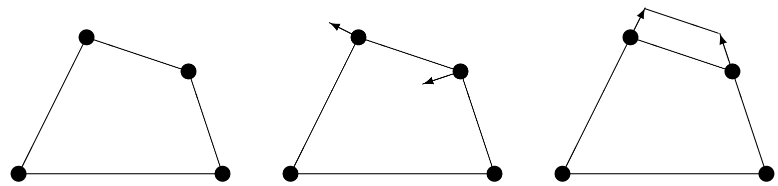


FIG. 2.1. *Infinitesimal motions and parallel redrawings.*

TABLE 2.1
First order terminology.

Plane design	Bar frameworks	Parallel drawings
Locally unique solution	rigid	tight
Locally unique solution with independent constraints	isostatic	minimally tight
Infinitely many nontrivial solutions	flexible	loose

dilation or contraction of $G(\mathbf{p})$, and if $\mathbf{u}$ is a nontrivial infinitesimal motion, $G(\mathbf{p}+\mathbf{u}^{\perp})$ will not be similar to $G(\mathbf{p})$.

Conversely, every parallel redrawing of a framework in the plane induces an infinitesimal motion of the framework. More directly, given a graph $G = (V, E)$, we can interpret the edges as line segments in the plane whose direction is to be fixed and thereby obtain the theory of *parallel drawings*, or direction designs, which is equivalent to the linearized problem obtained from interpreting the edges of G as length constraints. In Table 2.1 we compare the corresponding terminology used in these two theories.

3. Direction-length designs. The equivalent theories of parallel drawings and infinitesimal analysis of frameworks make tractable plane designs of lengths alone, and directions alone. We now mix these two types of constraints into a single system with an inclusive theory of designs with both kinds of constraints.

To distinguish the two kinds of constraints in figures of designs, we will follow the convention of indicating a length constraint between two points as an ordinary edge, and a direction constraint between two points as an edge with two arrowheads along its interior (see Figure 3.1).

DEFINITION 3.1. *A direction-length design is a double graph $FG = (V; D, L)$, where D, L are two sets of edges (no loops), and an assignment $\mathbf{p}$ of points $\mathbf{p_i} \in \mathbb{R}^2$ for each vertex $i \in V$. We call elements of D direction constraints and elements of L length constraints. Together, these are written as the design $FG(\mathbf{p})$.*

The edges L represent pairs of points whose lengths are held fixed. The edges D represent pairs whose directions are fixed. Since D and L need not be disjoint, a particular pair may have both types of connections. We also speak of the *direction graph* $F = (V, D)$ and the *length graph* $G = (V, L)$. We say that a direction-length design is *pure* if it only has edges of one type, and *mixed* otherwise. Two direction-length designs are said to be equivalent if they differ by a translation (see Figure 3.1).

We recall that for lengths the first-order constraints on "infinitesimal motions" (derivatives of the point positions) are

$$(\mathbf{p}_i - \mathbf{p}_j) \cdot (\mathbf{u}_i - \mathbf{u}_j) = 0.$$

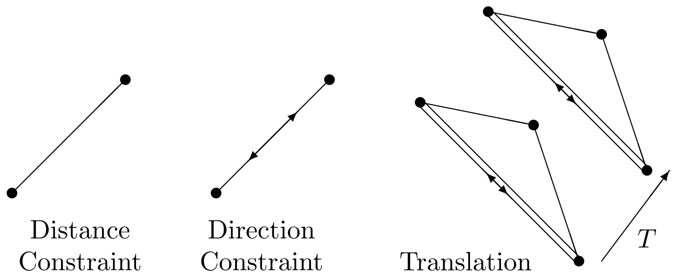


FIG. 3.1.

For plane directions, the constraint $(\mathbf{q}_i - \mathbf{q}_j) = \alpha(\mathbf{p}_i - \mathbf{p}_j)$ can also be rewritten in derivative form. The first step is to recall that the vector $(\mathbf{p}_i - \mathbf{p}_j)$ can be replaced by a constant normal $\mathbf{n}_{ij} = (\mathbf{p}_i - \mathbf{p}_j)^{\perp}$, and the equation becomes

$$\mathbf{n}_{ij} \cdot (\mathbf{p}(t)_i - \mathbf{p}(t)_j) = 0.$$

Taking derivatives, we obtain

$$\mathbf{n}_{ij} \cdot (\mathbf{u}_i - \mathbf{u}_j) = 0,$$

or equivalently

$$(\mathbf{p}_i - \mathbf{p}_j)^{\perp} \cdot (\mathbf{u}_i - \mathbf{u}_j) = 0.$$

Together, these produce a homogeneous linear system $R(FG, \mathbf{p}) \times \mathbf{u} = \mathbf{0}$. The matrix $R(FG, \mathbf{p})$ is the *constraint matrix* of the design. A set of constraints is *independent* if the corresponding rows of the matrix are independent. A solution to this system of constraints is called a *shake*. The design (with distinct vertices) is *stiff* if and only if this system has only the translations as solutions. Otherwise it is *shaky*. A set of constraints is *spanning* on the configuration $\mathbf{p}$ if it creates a stiff subdesign on these points. Equivalently, a spanning set of constraints spans the row space for the complete design on the configuration $\mathbf{p}$, with the complete graph on these vertices as both length and direction constraints.

Example 1. Consider the simple design $FG = (\{1,2\}; \{(1,2)\}, \{(1,2)\})$. The equations

$$|\mathbf{q}_1 - \mathbf{q}_2| = |\mathbf{p}_1 - \mathbf{p}_2| \text{ and } \mathbf{q}_1 - \mathbf{q}_2 = \alpha(\mathbf{p}_1 - \mathbf{p}_2)$$

are equivalent to the matrix equation

$$\begin{bmatrix} x_1 - x_2 & y_1 - y_2 & x_2 - x_1 & y_2 - y_1 \\ y_2 - y_1 & x_1 - x_2 & y_1 - y_2 & x_2 - x_1 \end{bmatrix} \times \begin{bmatrix} u_1 \\ w_1 \\ u_2 \\ w_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

If the points are distinct, it is easy to see that this system reduces to

$$\begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \times \begin{bmatrix} u_1 \\ w_1 \\ u_2 \\ w_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

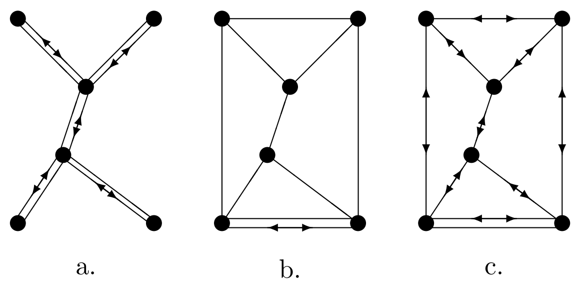


FIG. 3.2. *Three robust designs on* 6 *points.*

Thus $u_1 = u_2$ and $w_1 = w_2$, so the infinitesimal translation (u_1, w_1) is the only solution.

We are essentially interested in the rank (and independence) of the constraint matrix. The rank of the constraint matrix depends on both the double graph FG and the configuration $\mathbf{p}$. However, all generic $\mathbf{p}$ give the same rank for $R(FG, \mathbf{p})$, maximal over all configurations. A set of constraints is *generically independent* if it is independent for some (hence all) generic configurations. A set of edges is *generically spanning* if it is spanning for some (hence all) generic configurations.

Since any nonempty design has a two-dimensional space of translations in the plane, the maximum rank that the matrix can have is $2|V|-2$. A unique solution will therefore require $2|V|-2$ independent constraints, or equivalently $2|V|-2$ spanning constraints. Such sets, which are independent and spanning, induce a *robust design*. We may observe the following.

LEMMA 3.2.

1. *An independent set of* $|L| = 2|V| - 3$ *lengths plus any single direction constraint is an independent set of* $2|V| - 2$ *constraints; see Figure* 3.2b.
2. *An independent set of* $|D| = 2|V| - 3$ *directions plus any single length constraint is an independent set of* $2|V| - 2$ *constraints; see Figure* 3.2c.
3. *A spanning tree, used once as* L *for lengths and a second time as* D *for directions, is a spanning set of* $2|V| - 2$ *constraints; see Figure* 3.2a.
4. *If there are only length constraints, then every infinitesimal rotation is a shake.*
5. *If there are only direction constraints, then any infinitesimal dilation is a shake.*
6. *A spanning set of constraints must contain both direction and length constraints.*

3.1. Swapping. The form of the constraint matrix implies that lengths and directions play symmetric roles in the theory. In fact, we have a basic "duality" between these two constraints.

DEFINITION 3.3. *Given a double graph* $FG = (V; D, L)$, *the swapped double graph is* $FG^s = GF = (V; L, D)$, *where the roles of lengths and directions have been switched.*

In Figure 3.2a the swapped design is identical to the original, while b swaps to c.

THEOREM 3.4 (swapping theorem). *A direction-length design* $FG(\mathbf{p})$ *and the swapped design (the swapped double graph at the same points)* $FG^s(\mathbf{p}) = GF(\mathbf{p})$ *have isomorphic solution spaces of shakes.*

In particular, a direction-length design $FG(\mathbf{p})$ *is stiff (robust) if and only if the swapped design* $GF(\mathbf{p})$ *is stiff (robust).*

Proof. Consider the constraint matrix $R(FG, \mathbf{p})$ for the first design. If we rotate the design 90 degrees clockwise to form $\mathbf{q}$, then the independence of any set of constraints is unchanged and the matrices $R(FG, \mathbf{q})$ and $R(FG^s, \mathbf{p})$ are identical up to the sign of the rows. ∎

4. Robust designs. If a direction-length design has $2|V| - 2$ independent constraints, then the design is stiff, and the removal of any constraint introduces a shake. We called such a design robust. If a double graph FG has a configuration $\mathbf{p}$ for which the design $FG(\mathbf{p})$ is robust, we say the FG is *robust.* Equivalently, FG is robust if $FG(\mathbf{p})$ is robust for all generic configurations $\mathbf{p}$.

The term robust is used to indicate that small changes in the parameters of a design yield a "nearby" design with identical stiffness properties, which is highly desirable for ease of rendering and computability. This is indeed the case for robust double graphs, since the generic configurations comprise an open dense set of configurations.

5. Limiting designs. For our analysis, it is useful to expand the allowable designs to include typical limiting cases. For a given direction-length design $FG(\mathbf{p})$, the *normalized constraint matrix*, $R_n(FG, \mathbf{p})$, is obtained from $R(FG, \mathbf{p})$ by scaling the rows; multiplying row (i, j) by $|\mathbf{p}_i - \mathbf{p}_j|^{-1}$. The advantage of the normalized constraint matrix is that it has the same row dependencies as the original matrix, while its entries remain finite and nonzero under the limits $\lim_{\mathbf{p}_i \to \infty}$ and $\lim_{\mathbf{p}_i \to \mathbf{p}_j}$.

5.1. Vertices at infinity. Let $\mathbf{p}$ be a configuration of FG, and consider the limit of $R_n(FG, \mathbf{p})$ as $\mathbf{p}_i \xrightarrow{\mathbf{q}} \infty$ in the direction of a unit vector $\mathbf{q}$. Then the limit of a row corresponding to length constraint $l(i, j)$ of R_n has entries $\mathbf{q}$ in the columns corresponding to i, and $-\mathbf{q}$ in the columns corresponding to j, and the limit of a row corresponding to direction constraint $d(i, j)$ of R_n has entries $\mathbf{q}^\perp$ in the columns corresponding to i, and $-\mathbf{q}^\perp$ in the columns corresponding to j.

If the vertex i has two distinct neighbors, then $\lim_{\mathbf{p}_i \xrightarrow{\mathbf{q}} \infty} R_n(FG, \mathbf{p})$ is not the constraint matrix of a direction-length design, since the vertex i has no possible location. We will indicate a vertex at infinity as in Figure 5.1.

As a vertex tends to infinity, the edges in its star tend to parallelism, and so if a vertex has only direction constraints or only length constraints, then the limiting design has an infinitesimal motion even if none of the ordinary direction-length designs of the configuration do.

Example 2. Suppose we consider the complete graph on four vertices, $\mathbf{p}_0 = (-1, -1)$, $\mathbf{p}_1 = (+1, -1)$, $\mathbf{p}_2 = (0, 0)$, and $\mathbf{p}_3 = (0, 1)$ (see Figure 5.1a). The constraint matrix is

$$R(FG, \mathbf{p}) = \begin{bmatrix} 0 & -2 & 0 & 2 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 \\ -1 & -2 & 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & -2 & 0 & 0 & -1 & 2 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$

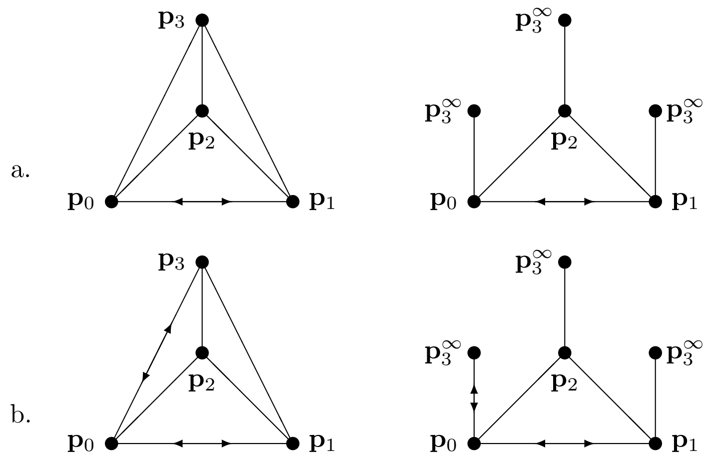


FIG. 5.1. *A point passing to infinity.*

and the normalized matrix is

$$R_n(FG, \mathbf{p}) = \begin{bmatrix} 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ -\alpha & -\alpha & 0 & 0 & \alpha & \alpha & 0 & 0 \\ 0 & 0 & \alpha & -\alpha & -\alpha & \alpha & 0 & 0 \\ -\beta & -2\beta & 0 & 0 & 0 & 0 & \beta & 2\beta \\ 0 & 0 & \beta & -2\beta & 0 & 0 & -\beta & 2\beta \\ 0 & 0 & 0 & 0 & 0 & -\alpha & 0 & \alpha \end{bmatrix},$$

where $\alpha = \frac{1}{\sqrt{2}}$ and $\beta = \frac{1}{\sqrt{5}}$). The limit as $\mathbf{p}_3 \xrightarrow{\mathbf{q}} \infty$, $\mathbf{q} = (0, 1)$, is the limit design on the right, with normalized matrix

$$\lim_{\mathbf{p}_3 \xrightarrow{\mathbf{q}} \infty} R_n(FG, \mathbf{p}) = \begin{bmatrix} 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ -\alpha & -\alpha & 0 & 0 & \alpha & \alpha & 0 & 0 \\ 0 & 0 & \alpha & -\alpha & -\alpha & \alpha & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix} \begin{matrix} 1 \\ \\ \\ -1 \\ 1 \\ \\ \end{matrix}$$

and infinitesimal motion $\mathbf{u}_3 = (1, 0)$ and $\mathbf{u}_0 = \mathbf{u}_1 = \mathbf{u}_2 = (0, 0)$. The numbers to the right of the matrix indicate the coefficients of a linear dependence of the rows.

The normalized matrix of the limit design in Figure 5.1b is

$$\lim_{\mathbf{p}_3 \xrightarrow{\mathbf{q}} \infty} R_n(FG, \mathbf{p}) = \begin{bmatrix} 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ -\alpha & -\alpha & 0 & 0 & \alpha & \alpha & 0 & 0 \\ 0 & 0 & \alpha & -\alpha & -\alpha & \alpha & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix},$$

which allows no nontrivial motion.

Since the limit of a dependent set is a dependent set in the limit design, an independent set in the limit design implies the nearby regular designs are also independent. If the limit design is spanning, then the nearby designs are also spanning.

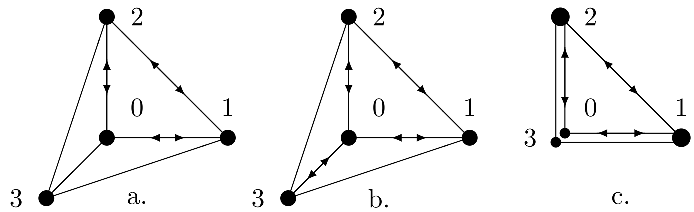


FIG. 5.2.

5.2. Infinitesimal edges. The points in a direction-length design are assumed to be distinct. However, it is sometimes useful to consider the limit design as one point $\mathbf{p}_i$ approaches another point $\mathbf{p}_j$ in the direction of the unit vector $\mathbf{q}$. The row for a length constraint $l(i,j)$ in the limit of the normalized constraint matrix will have $\mathbf{q}$ in the columns corresponding to i and $-\mathbf{q}$ in the columns corresponding to j. The row for a direction constraint $d(i,j)$ in the limit of the normalized constraint matrix will have $\mathbf{q}^\perp$ in the columns corresponding to i and $-\mathbf{q}^\perp$ in the columns corresponding to j.

Example 3. Consider the designs of Figure 5.2a and b. It is straightforward to check that both designs are generically independent.

If we take the limit as $\mathbf{p}_3$ approaches $\mathbf{p}_0$ along the direction $(1,1)$, Figure 5.2c, then the limit of design 5.2a has matrix

$$\begin{bmatrix} 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & \alpha & \alpha & -\alpha & -\alpha & 0 & 0 \\ \alpha & \alpha & 0 & 0 & 0 & 0 & -\alpha & -\alpha \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix},$$

which has rank 6, while the limit of design 5.2b has matrix

$$\begin{bmatrix} 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & \alpha & \alpha & -\alpha & -\alpha & 0 & 0 \\ -\alpha & \alpha & 0 & 0 & 0 & 0 & \alpha & -\alpha \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \begin{matrix} \alpha \\ \alpha \\ -1 \\ 1 \\ \alpha \\ -\alpha \end{matrix},$$

which has rank 5, ($\alpha = \frac{1}{\sqrt{2}}$).

Again, the limit of a dependent set is a dependent set in the limit design and an independent (spanning) set in the limit design implies the nearby regular designs are also independent (spanning).

5.3. Cycles on 3 vertices. In this section we describe small cycles which will be useful in subsequent arguments.

A *cycle* is a minimally dependent set of constraints. Among 3 vertices any set of 5 constraints is dependent, so the designs of Figures 5.3a and 5.3b are dependent. To see they are cycles, we need only observe that removing any constraint yields a robust design. These are both *generic cycles.* We can have a cycle on fewer than 5 edges if the position is special.

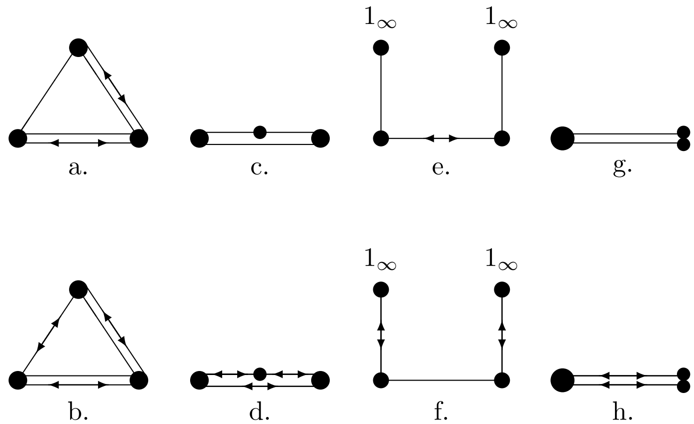


Fig. 5.3. *Cycles with* 3 *vertices.*

The design of Figure 5.3c is clearly a cycle, with matrix

$$\begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 2 & 0 & 0 & 0 & -2 & 0 \end{bmatrix} \begin{matrix} 2 \\ 2 \\ -1 \end{matrix}$$

and dependence given in the right column, similarly for Figure 5.3d.

The design of Figure 5.3e has point 1 approach ∞ in the vertical direction. The matrix is

$$\begin{bmatrix} 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \begin{matrix} 1 \\ -1 \\ 1 \end{matrix}$$

and similarly for Figure 5.3f.

Last, the design of Figure 5.3g has point 1 approach, point 2 in the vertical direction $(0,1)$ with the direction edge $d(1,2)$. The matrix is

$$\begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \end{bmatrix} \begin{matrix} 1 \\ -1 \\ 1 \end{matrix}$$

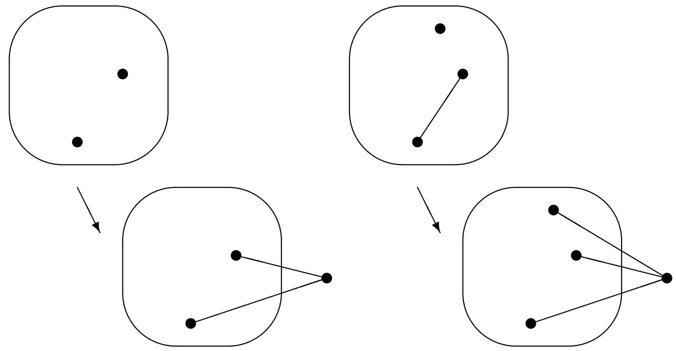
and similarly for Figure 5.3h with vertical length edge $l(1,2)$.

6. Extendability. For plane rigidity and plane directions, the simple inductive constructions for the independent (rigid) structures are the oldest characterizations (see [5, 16, 20]). In the proof of our broader combinatorial characterization, an inductive construction for robust direction-length designs remains a key step.

6.1. 0-extensions.

Definition 6.1. *Let $FG = (V; D, L)$ be a double graph. Let FG' be the double graph obtained from FG by adjoining a new vertex v whose total degree is* 2. *We say that FG' is a* 0*-extension of FG (see Figure* 6.1a*).*

The neighbors of the new vertex v need not be distinct vertices if the two new constraints at v are of a different type.



a. b.

FIG. 6.1. *Extensions.*

Let FG' be a 0-extension of FG and let v be the new vertex. Then the matrix of FG' is in block form

$$\left[\begin{array}{c|cc} A & \multicolumn{2}{c}{\mathbf{0}} \\ \hline B & (x_v - x_a) & (y_v - y_a) \\ & (x_v - x_b) & (y_v - y_b) \end{array}\right]$$

if the new edges are both lengths,

$$\left[\begin{array}{c|cc} A & \multicolumn{2}{c}{\mathbf{0}} \\ \hline B & -(y_v - y_a) & (x_v - x_a) \\ & -(y_v - y_b) & (x_v - x_b) \end{array}\right]$$

if they are both directions, and

$$\left[\begin{array}{c|cc} A & \multicolumn{2}{c}{\mathbf{0}} \\ \hline B & (x_v - x_a) & (y_v - y_a) \\ & -(y_v - y_b) & (x_v - x_b) \end{array}\right]$$

if there is one of each type. So the rows corresponding to the new constraints in the new matrix are independent of the other rows if the new edges are not parallel, in the first two cases, or perpendicular, in the third case, and we have the following.

LEMMA 6.2. *Let FG' be a 0-extension of FG and suppose FG is independent with respect to some configuration $\mathbf{p}$. Then $\mathbf{p}$ may be extended to the new vertex so that FG' is also independent.*

In particular, if FG is generically independent or robust, then any 0-extension of FG is generically independent or robust, respectively.

6.2. 1-extensions.

DEFINITION 6.3. *Let $FG = (V; D, L)$ be a double graph with edge f. A 1-extension of FG, FG', is obtained from FG by removing the edge f and adding a new vertex v of degree 3 so that*

1. *the neighbors of v include both endpoints of f,*
2. *neither D nor L decrease in size.*

We can think of the new edges (v, a) and (v, b) as *splitting* the constraint $l(a, b)$ or $d(i, j)$ (see Figure 6.1b). Condition 2 is satisfied as long as a length constraint is not replaced by three direction constraints, or vice versa. A configuration is *general* if no three points are collinear.

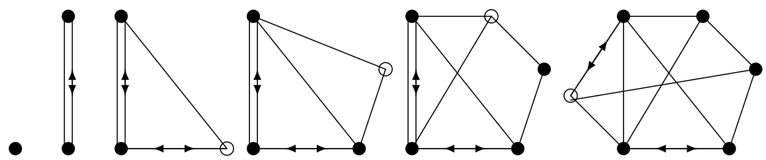


FIG. 6.2. *A direction-length construction.*

LEMMA 6.4. *Let FG' be a 1-extension of FG, and let FG be independent (spanning) with respect to a general configuration $\mathbf{p}$. Then $\mathbf{p}$ can be extended so that FG' is also independent (spanning).*

Proof. By the swapping theorem (Theorem 3.4), we assume without loss of generality (w.l.o.g.) that $f \in L$.

Let $\{a, b\}$ be the endpoints of f and let v denote the new vertex with new edges (v, a), (v, b), and (v, c).

Case 1. Let $l(v, a), l(v, b) \in L$, c distinct from a and b. We can adjoin v by a 0-extension to vertices a and c with constraints $l(v, a), l(v, b)$ with the new vertex v placed along the segment from $\mathbf{p}_a$ to $\mathbf{p}_b$. Then, since a triangle of lengths with vertices on a line is a cycle, we can replace the constraint $l(a, b)$ with the constraint $l(v, b)$ so that FG' is independent (spanning).

Case 2. Let $l(v, a) \in L$ and $d(v, b) \in D$. We can adjoin v by a 0-extension with constraints $l(v, a)$ and $l(v, c)$ and take the limit $\mathbf{p}_a \xrightarrow{\mathbf{q}} \mathbf{p}_b$ in the direction $\mathbf{q}$ perpendicular to (a, b). Since the rows for $l(a, b)$, $l(v, a)$ form a cycle with the infinitesimal direction $d(v, b)$, we can replace $l(a, b)$ with $d(b, v)$ and the limiting design is independent (spanning). Therefore any nearby generic configuration gives an independent (spanning) design.

Case 3. Let $d(v, a), d(v, b) \in D$, c distinct from a and b. Then again adjoin v by a 0-extension, and let $\mathbf{v}$ approach ∞ in the direction $\mathbf{q}$ perpendicular to (a, b). In this position, the rows for $l(a, b)$, $d(v, a)$, and $d(v, b)$ form a cycle with (v, b), so we can replace $l(a, b)$ with $d(b, v)$ and the limiting design is independent (spanning). Again, any generic $\mathbf{p}$ is also independent (spanning). □

Remark. Notice that the "limiting design" argument does, indeed, break down if we try the forbidden replacements: replace a single direction with three lengths, or replace a single length by three directions. With a limiting point "at infinity," all three directions (or lengths) will be parallel rows of the matrix, and the initial 0-extension will fail to be independent.

6.3. Direction-length constructions. In the spirit of the classical Henneberg sequences, we now describe how to obtain complex robust designs from a single vertex using only the simple extensions just developed.

DEFINITION 6.5. *A direction-length construction of the double graph $FG = (V; D, L)$ is a sequence of direction-length double graphs,*

$$FG_1,\ FG_2, \ldots FG_{|V|},$$

beginning with the single vertex graph FG_1, ending with $FG_{|V|} = FG$, such that FG_k is a 0-extension or 1-extension of FG_{k-1} (see Figure 6.2).

From Lemmas 6.2 and 6.4 we have the following theorem.

THEOREM 6.6. *A double graph FG with a direction-length construction is generically robust.*

In section 7, the converse is demonstrated. Since the class of constructions is

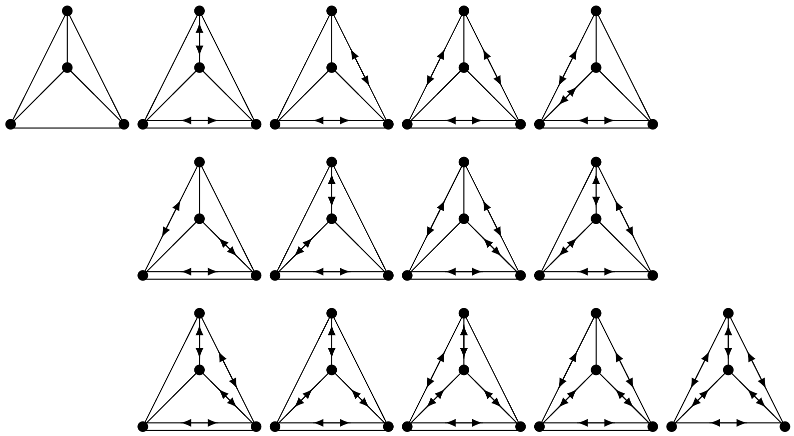


FIG. 6.3. *Generic 3-connected cycles on 4 vertices.*

closed under swapping, the class of constructible designs is closed under swapping.

6.4. Generic cycles on 4 vertices. Let us enumerate the generic cycles on 4 vertices, that is, those double graphs whose edges correspond to minimally dependent sets of constraints.

A generic cycle cannot have a vertex of total valence 2 (or less) since that would be a 0-extension of an independent set, or a 0-extension of a smaller cycle. On the other hand, on 4 vertices, a set of 6 directions or 6 lengths must be dependent, as well as a set of 7 edges of mixed type. Thus a cycle on 4 vertices is either

1. a tetrahedron of lengths;
2. a tetrahedron of directions;
3. the edges of both types form a tetrahedron with a doubled edge (the graph is vertex 3-connected);
4. the edges of both types form two attached triangles, with a doubled edge in each (not the shared edge) (the graph is vertex 2-connected).

Moreover, the third type must have at least 2 edges of each kind, since if there was only one, then deleting it would leave a pure tetrahedron which is dependent. Also cycles of type 4 must have at least two edges of each kind, since there is a pair of doubled edges.

All candidates of types 1–3 are listed in Figure 6.3. To see that the mixed graphs are all in fact generic cycles, one may easily give a direction-length construction for each of the graphs with any one edge deleted.

The circuits of type 4 can be constructed from two of the cycles on three vertices by cycle exchange. Figure 6.4 illustrates this process. The single lines represent constraints of either kind, while the double lines indicate that constraints of both kinds are present. Altogether there are 12 circuits of type 4.

With the exception of the 2 pure cycles, all generic cycles on 4 vertices may be obtained from the generic cycles on 3 vertices by either 1-extension or cycle-exchange. It seems plausible that all generic cycles may be obtained from the generic cycles on 3 and 4 vertices by a sequence of extensions and cycle exchanges, but to date no proof is known, not even in the case of pure designs.

7. The generic matroid. Consider a complete double graph $K_n^2 = (V; D_c, L_c)$ on $V = \{1, \ldots, n\}$ together with a generic configuration $\mathbf{p}$. Since we are interested

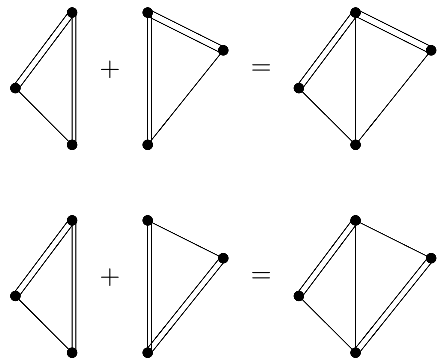


FIG. 6.4. *Generic* 2*-connected cycles on* 4 *vertices.*

in the combinatorial properties of the matrix $R(K_n^2) = R(K_n^2, \mathbf{p})$, we examine the matroid $\mathsf{CAD}_{\mathsf{dl}}(n)$ defined by the rows of $R(K_n^2)$, which we call the *generic* dl*-cadroid on* n *vertices.* Theorem 6.6 states that every double graph on n vertices with a direction-length construction is a basis of $\mathsf{CAD}_{\mathsf{dl}}(n)$.

We know that the rank of the full constraint matrix for a generic configuration of n points in $\mathbb{R}^2$ has rank $2n-2$. Also, for all $k < n$, $\mathsf{CAD}_{\mathsf{dl}}(k)$ may be viewed as a restriction of $\mathsf{CAD}_{\mathsf{dl}}(n)$. Therefore we can offer clear necessary conditions for a basis B of $\mathsf{CAD}_{\mathsf{dl}}(n)$.

$\mathsf{CAD}_{\mathsf{dl}}1$: $|B| = 2n-2$;

$\mathsf{CAD}_{\mathsf{dl}}2$: for all nonempty subsets $E \subseteq B$

$$|E| \leq 2|V(E)| - 2;$$

$\mathsf{CAD}_{\mathsf{dl}}3$: for all pure nonempty subsets $E \subseteq B$,

$$|E| \leq 2|V(E)| - 3.$$

Theorem 7.4 will show that these are also sufficient.

We first show that $\mathsf{CAD}_{\mathsf{dl}}1, \ldots, \mathsf{CAD}_{\mathsf{dl}}3$ define the bases of a matroid Count(n) on $D_c \cup L_c$ and then show that this matroid is isomorphic to $\mathsf{CAD}_{\mathsf{dl}}(n)$.

THEOREM 7.1. *Let* $K^2 = (V; L_c, D_c)$ *denote the complete double graph on* $|V|$ *vertices. Then the collection of subsets* $B \subseteq L_c \cup D_c$ *which satisfy* $\mathsf{CAD}_{\mathsf{dl}}1, \ldots, \mathsf{CAD}_{\mathsf{dl}}3$ *are the bases of a matroid on* $L_c \cup D_c$.

Proof. We will show that the collection $\mathcal{C}$ of minimal sets which violate $\mathsf{CAD}_{\mathsf{dl}}1$, $\ldots$, $\mathsf{CAD}_{\mathsf{dl}}3$ satisfy the cycle axioms for a matroid.

If $C \in \mathcal{C}$ is pure, then $|C| = 2|V(C)| - 2$ and $|C'| \leq 2|V(C')| - 3$ for all proper nonempty subsets C' of C.

If $C \in \mathcal{C}$ and C is mixed, then C must contain at least two elements from both D_c and L_c. We have $|C| = 2|V(C)| - 1$ and all proper subsets of C must be independent, i.e., satisfy $\mathsf{CAD}_{\mathsf{dl}}3$.

We need to show that if $C_1, C_2 \in \mathcal{C}$, and $x \in C_1 \cap C_2$, then there exists $C_3 \in \mathcal{C}$, $C_3 \subseteq C_1 \cup C_2 - x$.

Let the supports of C_1 and C_2 have cardinalities m and n, respectively, and let the support of $C_1 \cap C_2$ be i.

If C_1 and C_2 are both mixed, then we have

$$|C_1 \cup C_2 - e| = |C_1| + |C_2| - |C_1 \cap C_2| - 1$$

$$\geq 2n - 1 + 2m - 1 - (2i - 2) - 1 = 2(m + n - i) - 1$$
$$= 2|V(C_1 \cup C_2)| - 1,$$

so $C_1 \cup C_2 - e$ contains an element of $\mathcal{C}$ since it violates $\mathsf{CAD}_{\mathsf{dl}}2$.

If C_1 is mixed and C_2 is pure, then their intersection has at most $2i - 3$ edges and C_2 also has one edge fewer than before, so we arrive at the following conclusion:

$$|C_1 \cup C_2 - e| = 2(m + n - i) - 1 = 2|V(C_1 \cup C_2)| - 1.$$

If C_1 and C_2 are both pure (of the same type, since they have nonempty intersection), then

$$|C_1 \cup C_2 - e| = 2(m + n - i) - 2 = 2|V(C_1 \cup C_2)| - 2.$$

Since $C_1 \cup C_2$ is also pure, this gives the dependence. □

This result is a particular case of a more general construction of matroids from "submodular counts" described in [23].

Observe that the generic cycles of $\mathsf{CAD}_{\mathsf{dl}}(n)$ listed in Figures 5.3 and 6.3 are also cycles in Count(n) and these cycles are in fact all the cycles of Count(n) on 3 or 4 vertices. Notice also that the symmetry of the definition of Count(n) directly demonstrates the invariance of all matroidal properties under swapping.

We need the following lemmas.

LEMMA 7.2. *If B is a basis of Count(n), then the double graph induced by B is edge 2-connected.*

Proof. If $B - e$ is disconnected with two components on k and l vertices, then the rank of B is at most $2(k + l) - 3$. □

LEMMA 7.3. *Let I be independent in a matroid and let C be a cycle in this matroid. Then for each element $e \in I \cap C$ there is an element $f \in C - I$ so that $I - e + f$ is independent.*

Proof. Let $e \in I \cap C$. Assume that for each $f \in C - I$, $I - e + f$ is dependent. Then $C - e$ is a subset of the closure of $I - e$. Since e is in the closure of $C - e$, e is in the closure of $I - e$. Since $e \in I$ and I is independent, this is a contradiction. □

THEOREM 7.4. *For any set B of edges in K_n^2 the following are equivalent:*

1. *B is a basis of Count(n);*
2. *B is a basis of $\mathsf{CAD}_{\mathsf{dl}}(n)$;*
3. *B has a direction-length construction.*

Proof. (3) ⇒ (2). By Theorem 6.6, every set with a direction-length construction is a basis of $\mathsf{CAD}_{\mathsf{dl}}(n)$.

(2) ⇒ (1). Every basis of $\mathsf{CAD}_{\mathsf{dl}}(n)$ satisfies CAD1, …, CAD3 and so is a basis for Count(n).

(1) ⇒ (3). The proof is by induction on the number of vertices. It is trivial for 2 vertices.

Assume it is true for $n-1$ vertices. Since the average valence of a basis in Count(n) is $4(1 - 1/n) < 4$, there is some vertex of total valence ≤ 3. By the 2-connectivity, this vertex must have valence either 2 or 3. If the valence is 2, then the robust set is the 0-extension of a smaller independent set, and we are done.

Assume B has a vertex v of valence 3. If star(v), the set of constraints with endpoint v, is mixed (has constraints of both types), we add constraints among the neighbors of v to create a Count(k), $k = 2$ or 3, basis B_v for these neighbors. Adding the three valent vertex v, we have a dependent set in Count(n) and therefore a small cycle C containing v. We have $C \not\subseteq B$, but star(v) $\subseteq C \cap B$. By Lemma 7.3, for any edge e in star(v), there is an $f \in C - B$ such that $B - e + f$ is independent, in fact, a basis, B'. Therefore, $B_{n-1} = B' - \text{star}(v)$ is a basis of Count($n-1$), and by induction it has a construction. Since every replacement of a constraint f by a mixed vertex is a valid 1-extension, B is a 1-extension of B_{n-1}. Therefore B has a construction.

If star(v) is pure (say all lengths up to swapping), then it has 3 distinct neighbors. Adding length constraints among these neighbors will produce a unique pure cycle C – the complete graph on 4 vertices. As before, for any edge $e \in \text{star}(v)$, we can find a length constraint $f \in C - B$ such that $B - e + f$ is independent. The replacement of a length f in B_{n-1} by the 3-lengths at v is a valid 1-extension.

This completes the induction. □

Remark. The characterization of Count(n), by the count, appears to be exponential: "for all subsets B' … ." However, by a theorem of Nash-Williams [10, 11], independent sets are decomposable into two spanning forests with the additional condition that no two subtrees that both contain only edges of D or edges of L do not have the same span. A general matroidal algorithm by Edmonds can be used to provide such a decomposition in polynomial time. Also Sugihara [15] and Imai [6] have general polynomial time algorithms to verify such conditions.

For length designs (and therefore also direction designs) Crapo has adapted Edmonds's algorithm to also give a low degree polynomial algorithm for the tree structures which correspond to the counts $2|V(E)| - 3$. It is clear that this approach could be modified for our closely related counts, giving polynomial time algorithms to confirm a basis (or extract a basis from a spanning set). This algorithm would have the additional advantage that its output (the two trees mentioned above) could be displayed for rapid visual verification.

Remark. There are some additional results on both necessary and sufficient connectivity for spanning sets. All of these results are, in some form, the direct analogues of results for length designs (plane frameworks). All of the proofs are based on the counting properties of Count(n).

1. Circuits in $\mathsf{CAD}_{\mathsf{dl}}(n)$ are vertex 2-connected and edge 3-connected.
2. All circuits of $\mathsf{CAD}_{\mathsf{dl}}(n)$ are spanning on their vertices.
3. If a direction-length design is vertex 6-connected and mixed, then it is spanning. This is a direct analogue of a result of [9] for frameworks. Their proof (also based on counts) extends with small modifications.

In the 5-connected 5-regular frameworks example of Lovasz and Yemini [9], we can double one of the 5-cliques to get an example of a 5-connected double graph which is not stiff.

8. Concluding remarks. Our entire analysis of constraints has been "local," with robustness guaranteeing local uniqueness for *small* changes in the configuration, up to congruence. As we mentioned in the introduction, the problem of global uniqueness up to congruence, for all configurations is more difficult. This is no longer a matter of linear algebra and matroids; it is quadratic algebra with all the attendant difficulties. For frameworks, this global uniqueness is called "global rigidity" [1]. For pure lengths, any basis of the generic rigidity matroid will *not* be globally rigid, except in special singular (nongeneric) configurations, where the design is dependent [4].

On the other hand, for pure directions, both global and local transformations are described by linear equations, and the design is globally unique, up to translations and dilations, if and only if it is locally unique.

For direction-length designs, we have both types of cases.

1. A robust direction-length design with one length and $2n - 3$ directions will

be globally unique, up to translation and dilation by -1, if and only if it is locally unique.

2. A direction-length design with one direction and the remaining constraints lengths will be globally unique, up to translation, dilation by -1, and reflection in the line of the single direction, if and only if the length design is globally unique, up to congruence.
3. A direction-length design which is globally unique, up to translation and dilation by -1, is 2-connected in a vertex sense. (Otherwise, we can take the point of disconnection, and dilate one of the components by -1 in this center.)

An inspection of a result and proof of Hendrickson [4] indicates that the following result also holds.

PROPOSITION 8.1. *A robust direction-length design FG with more than one length is not globally unique.*

As we mentioned in the introduction, our work with lengths and directions was motivated by a broader unsolved problem in plane CAD. Consider a design constrained by lengths between pairs of points and angles between lines. This angle constraint could involve two edges sharing a vertex or simply be the angle between to disjoint edges ("the following two lines are parallel"). The problem of a polynomial time algorithm, or direct combinatorial algorithm, for these constraints is unsolved and difficult [24]. (We do have the corresponding constraint matrices (which have nonzero vector entries under up to four vertices per angle row). By using variables for the coordinates of points, we have a well-defined generic matroid for the constraints $\mathsf{CAD}_{\mathsf{dl}}(n)$. Taking determinants, we get a superexponential "combinatorial" algorithm to check for bases, or independence in $\mathsf{CAD}_{\mathsf{dl}}(n)$.)

Writing A for the set of angle constraints (actually partially ordered triples and quadruples), and L for the length constraints, there is a necessary set of counts for B to be a basis of the matroid $\mathsf{CAD}_{sfda}(n)$:

$\mathsf{CAD}_{sfda}1$: $|B| = 2n - 3$;

$\mathsf{CAD}_{sfda}2$: for all nonempty subsets $E \subseteq B$

$$|E| \leq 2|V(E)| - 3;$$

$\mathsf{CAD}_{sfda}3$: for all nonempty subsets of angles $E \subseteq B$, $E \subseteq A$

$$|E| \leq 2|V(E)| - 4.$$

The subtracted constant 3 in $\mathsf{CAD}_{sfda}(n)$ corresponds to the translations and rotations of a robust design. The subtracted constant 4 in $\mathsf{CAD}_{sfda}3$ corresponds to the translations, rotations, and dilation permitted by a maximal pure angle design.

However, these conditions are not enough: any "polygon of angles" will be dependent, and in a quadrilateral, these four angles on four points will not violate the condition $\mathsf{CAD}_{sfda}3$. Even if we carefully insert this "polygon condition" (by adding variables directly for the edges, etc.) the added count will not be sufficient to define a matroid (as occurred for Count(n)). In practice, the appearance of such "nonspanning" circuits is a sign that the techniques employed in this paper, adapted from the study of plane frameworks, will be inadequate.

However, if we have an angle design in which the angles are linked together as a connected set among the attached edges (ideally a tree since any polygon is dependent; see Figure 8.1a), the design can be analyzed with our theory. Taking any one of the edges in these angles, and defining an arbitrary direction to it, we can work through the

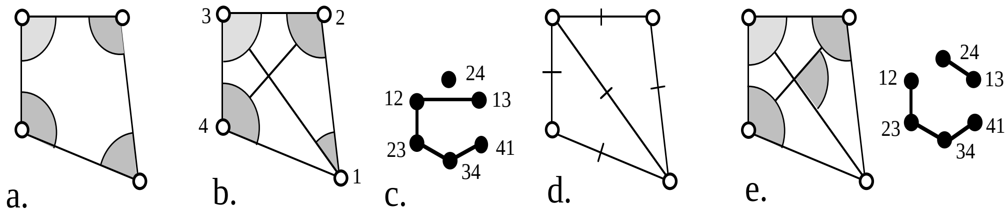


FIG. 8.1. *Generic cycles with angles.*

attached angles to assign a direction constraint for each of the angle constraints. This induces a direction-length design whose properties of robustness, independence, etc. directly correspond to the robustness, independence, etc. of the original angle-length design. The reader can check that, with one added direction and each angle converted to a direction constraint, the three conditions $\mathsf{CAD}_{sfda}1$, $\mathsf{CAD}_{sfda}2$, $\mathsf{CAD}_{sfda}3$ convert to the axioms for Count(n). We have solved this special case of the general unsolved problem of angle-length designs.

If the angles form a forest of several trees (see Figure 8.1), the combinatorial analysis becomes difficult and unsolved. One key difficulty is that we do not yet have an adequate list of inductive constructions which are guaranteed to generate all bases of the matroid $\mathsf{CAD}_{sfda}(n)$. Moreover, this list will have to involve inductive principles for vertices attached to up to 7 constraints, since each angle may involve up to four vertices. It is unclear whether there will be any polynomial time algorithm for general bases in $\mathsf{CAD}_{sfda}(n)$.

More generally, the lines could contain many points (not just two) and we would have additional incidence constraints for vertices lying on lines. This takes us into several other unsolved problems, both for incidences alone and for mixes of incidences, lengths, and angles [23].

Finally, we could convert "direction constraints" into directions for lines, but replace incidences with possibly nonzero distances from points to lines. Again certain special cases of this can be solved [14] and other extensions are unsolved.

We have focused on constraints in plane CAD because we have some substantial results. Many of the related problems in 3-space are substantially more difficult. For example, the problem of independent length constraints alone in 3-space is unsolved. While there is a corresponding matrix, and a partial list of inductive constructions, there is no combinatorial characterization (beyond the constraint matrix with variable entries and the associated superexponential algorithm).

For direction constraints in 3-space, there are substantial results. A "direction" for a line segment becomes two rows in the constraint matrix, corresponding to two planes, with assigned normals, containing the line. The entire theory of plane directions has an appropriate extension to this "polymatroid" (two rows for each edge). This approach is described in more detail in [17, 19, 24].

REFERENCES

[1] R. CONNELLY, *On generic global rigidity*, in Applied Geometry and Discrete Mathematics, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 4, AMS, Providence, RI, 1990, pp. 147–155.

[2] J. GRAVER, B. SERVATIUS, AND H. SERVATIUS, *Combinatorial Rigidity*, AMS, Providence, RI, 1993.

[3] J. Graver, B. Servatius, and H. Servatius, *Abstract rigidity in n-space,* Contemp. Math., AMS, 178 (1994), pp. 145–151.

[4] B. Hendrickson, *Conditions for unique graph realizations,* SIAM J. Comput., 21 (1992), pp. 65–84.

[5] L. Henneberg, *Die graphische Statik der starren Systeme*, Leipzig 1911, Johnson Reprint, 1968.

[6] M. Imai, *A double-tree structured multicomputer system and its application to combinatorial problems,* Trans. IEICE, E69, 9 (1986), pp. 1002–1010.

[7] G. Kramer, *Solving Geometric Constraint Systems (A case study in kinematics)*, MIT Press, Cambridge, MA, 1992.

[8] G. Laman, *On graphs and the rigidity of plane skeletal structures*, J. Engrg. Math., 4 (1970), pp. 331–340.

[9] L. Lovasz and Y. Yemini *On generic rigidity in the plane*, SIAM J. Alg. Discrete Methods, 3 (1982), pp. 91–98.

[10] C. St. J. A. Nash-Williams, *Edge disjoint spanning trees of finite graphs*, J. London Math. Soc., 36 (1961), pp. 445-450.

[11] C. St. J. A. Nash-Williams, *Decompositions of finite graphs in forests*, J. London Math. Soc., 39 (1964), p. 12.

[12] J. C. Owen, *Algebraic solutions for geometry from dimensional constraints*, in Symposium on Solid Modeling Foundations and CAD/CAM Applications, ACM Press, New York, 1991.

[13] J. C. Owen, *Constraints on Simple Geometry in Two and Three Dimensions*, preprint, D-Cubed Ltd., Cambridge, UK.

[14] J. C. Owen and W. Whiteley, *Constraining Plane Geometric Configurations in CAD: Directions and Distances*, preprint, Department of Mathematics and Statistics, York University, North York, ON.

[15] K. Sugihara, *Detection of structural inconsistency in systems of equations with degrees of freedom and its applications*, Discrete Appl. Math., 10 (1985), pp. 297–312.

[16] T.-S. Tay and W. Whiteley, *Generating all isostatic frameworks*, Structural Topology, 11 (1985), pp. 21–69.

[17] W. Whiteley, *Parallel Redrawing of Configurations in* 3*-Space*, preprint, Department of Mathematics and Statistics, York University, North York, ON, Canada.

[18] W. Whiteley, *The union of matroids and the rigidity of frameworks*, SIAM J. Algebraic Discrete Meth., 1 (1988), pp. 237–255.

[19] W. Whiteley, *Some matroids on hypergraphs with applications to scene analysis and geometry*, Discrete Comput. Geom., 4 (1988), pp. 75–95.

[20] W. Whiteley, *Vertex splitting in isostatic frameworks*, Structural Topology, 16 (1990), pp. 23–30.

[21] W. Whiteley, *Matroids and rigidity*, in Matroid Applications, N. White, ed., Cambridge University Press, London, Cambridge, 1993.

[22] W. Whiteley, *Constraining Plane Configurations in CAD: Geometry of Directions and Lengths*, preprint, Department of Mathematics and Statistics, York University, North York, ON, Canada.

[23] W. Whiteley, *Matroids for discrete applied geometry*, in Proc. AMS Conference on Matroids, Seattle, WA, Contemp. Math., J. Bonin, J. Oxley, and B. Servatius, eds., AMS, Providence, RI, 1995.

[24] W. Whiteley, *Constraining Plane Configurations in CAD: Angles*, preprint, Department of Mathematics and Statistics, York University, North York, ON, Canada.

SIAM J. DISCRETE MATH.
Vol. 12, No. 2, pp. 155–159

ON STIRLING NUMBERS FOR COMPLEX ARGUMENTS AND HANKEL CONTOURS*

PHILIPPE FLAJOLET† AND HELMUT PRODINGER‡

Abstract. Cauchy coefficient integrals and Hankel contours provide a natural generalization of Stirling numbers for unrestricted complex values of their arguments. Many classical identities survive such an extension.

Key words. Stirling numbers, complex arguments, Hankel contours

AMS subject classification. 05A10

PII. S0895480198332594

1. Introduction. Richmond and Merlini introduced in [5] an extension of Stirling's subset numbers $\left\{{x \atop y}\right\}$ and cycle numbers $\left[{x \atop y}\right]$ when $x-y$ is an integer. They also propose a further generalization when $x-y$ is not an integer, but most classical properties are no longer preserved. As the authors say about their most general extension in [5, p. 76]: "It seems to us that the ideas used to derive identities and recurrences lead to complicated formulas in general. There are significant terms resulting from the fact that the integrands are not single valued and also from the fact that the contours change."

In this note, we give an alternative and more natural extension of Stirling numbers of complex arguments for which most classical identities are still satisfied. (We restrict ourselves to the most common properties, leaving it to the imagination of the reader to go further.) As in [5], our approach starts with Cauchy coefficient integrals. However, in contrast to [5], we use a Hankel contour that has the merit of *not* being dependent on particular index values. This intrinsic character of the contour precisely ensures the permanence of identities.

Relevant references for the classical theory are [1, 2, 3, 4], the latter paper being an excellent historical account of Stirling numbers.

2. Stirling numbers of complex index. By definition, the Stirling subset numbers ("of the second kind") $\left\{{n \atop k}\right\}$ are for $n, k \in \mathbb{N}$ given by

$$\left\{{n \atop k}\right\} = \frac{n!}{k!}[z^n]\,(e^z-1)^k$$

or, by Cauchy's coefficient formula,

$$\left\{{n \atop k}\right\} = \frac{n!}{k!}\frac{1}{2i\pi}\int_\gamma (e^z-1)^k\,\frac{dz}{z^{n+1}}, \tag{1}$$

*Received by the editors January 16, 1998; accepted for publication September 22, 1998; published electronically April 29, 1999. This research was supported in part by the Long Term Research Project *Alcom-IT* 20244 of the European Union and by the French–Austrian Cooperation Project Amadée 11j 2868/1997.

http://www.siam.org/journals/sidma/12-2/33259.html

†Algorithms Project, INRIA, F-78150 Rocquencourt, France (Philippe.Flajolet@inria.fr, http://www-rocq.inria.fr/algo/flajolet/index.html).

‡Centre for Applicable Analysis and Number Theory, Mathematics Department, University of the Witwatersrand, 2050 Johannesburg, South Africa (helmut@gauss.cam.wits.ac.za, http://www.wits.ac.za/helmet/index.htm).

where the integration contour γ is a small contour encircling the origin. As n is nonnegative in (1), the contour γ can be deformed into a *Hankel contour* $\mathcal{H}$ (see [6]) that starts from $-\infty$ below the negative axis, surrounds the origin counterclockwise, and returns to $-\infty$ in the half plane $\Im z > 0$. Details of $\mathcal{H}$ are of course immaterial, and we need only assume that it is at distance ≤ 1 from the real axis.

This suggests the following definition.

DEFINITION 1. *The Stirling numbers of complex arguments ("fractional order") are defined for* $\Re(x) > 0$ *by*

$$\left\{ {x \atop y} \right\} = \frac{x!}{y!} \frac{1}{2i\pi} \int_{\mathcal{H}} (e^z - 1)^y \, \frac{dz}{z^{x+1}}, \tag{2}$$

where $s! = \Gamma(s+1)$*. The determination of* $(e^z - 1)^y$ *is the principal determination on the part of the contour* $\Re z > 0$ *extended by continuity to the whole of* $\mathcal{H}$*.*

This definition extends the Stirling subset numbers to *arbitrary* complex arguments (x, y) satisfying $\Re x > 0$. When $\Re(x) \leq 0$, the integral diverges. However, through integration by parts, one finds when $\Re(x) > 1$

$$\left\{ {x \atop y} \right\} = \frac{(x-1)!}{(y-1)!} \frac{1}{2i\pi} \int_{\mathcal{H}} e^z \, (e^z - 1)^{y-1} \, \frac{dz}{z^x}. \tag{3}$$

The integral in (3) now converges for all values of x and y. The variant form (3) shows that $\left\{ {x \atop y} \right\}$ can be continued for $\Re(x) \leq 0$ into a meromorphic function of x (for any fixed y), with poles at the nonpositive integers. As a function of y (for any fixed x not a negative integer), it is entire.

Our definition of generalized Stirling numbers in (2) and (3) coincides with that of [5] *only* when $x - y$ is an integer. It differs significantly in other cases, since Richmond and Merlini propose to define the general form of $\left\{ {x \atop y} \right\}$ by means of a saddle point circle that, contrary to $\mathcal{H}$, is dependent upon the particular values of x, y.

3. Relations. As announced, we show now that the most common properties are preserved for our generalized Stirling numbers as defined by (2) and (3).

Recurrence. In the integral representation for $\left\{ {x \atop y} \right\}$, perform integration by parts. This gives for $\Re x > 1$

$$\frac{1}{2i\pi} \int_{\mathcal{H}} (e^z - 1)^y \, \frac{dz}{z^{x+1}} = \left[-\frac{1}{xz^x} (e^z - 1)^y \right]_{\mathcal{H}} + \frac{y}{x} \frac{1}{2i\pi} \int_{\mathcal{H}} (e^z - 1)^{y-1} \, e^y \frac{dz}{z^x}$$

and upon writing $e^y = (e^y - 1) + 1$,

$$\frac{1}{2i\pi} \int_{\mathcal{H}} (e^z - 1)^y \, \frac{dz}{z^{x+1}} = \frac{y}{x} \frac{1}{2i\pi} \int_{\mathcal{H}} (e^z - 1)^y \, e^y \frac{dz}{z^x} + \frac{y}{x} \frac{1}{2i\pi} \int_{\mathcal{H}} (e^z - 1)^{y-1} \, e^y \frac{dz}{z^x}$$

or in standard notation,

$$\left\{ {x \atop y} \right\} = \left\{ {x-1 \atop y-1} \right\} + y \left\{ {x-1 \atop y} \right\}.$$

This relation originally established for $\Re x > 1$ persists for all complex x by uniqueness of analytic continuation.

Binomial formula. A binomial expansion of $(e^y-1)^k$ yields the classical formula $(n, k \in \mathbb{N})$,

$$\left\{ {n \atop k} \right\} = \frac{1}{k!} \sum_{j=1}^{k} \binom{k}{j} (-1)^{k-j} j^n.$$

This process naturally extends to complex x giving, for all $k \in \mathbb{N}$,

$$\left\{ {x \atop k} \right\} = \frac{1}{k!} \sum_{j=1}^{k} \binom{k}{j} (-1)^{k-j} j^x,$$

upon using the binomial expansion in (2) and appealing to Hankel's original representation of the gamma function [6].

Bell numbers. The Bell numbers of integral order are defined by their exponential generating function

$$\mathcal{B}_n = n![z^n]e^{e^z-1},$$

and they satisfy the relation

$$\mathcal{B}_n = \sum_{k=0}^{n} \left\{ {n \atop k} \right\}.$$

This suggests to define the Bell numbers of any complex order x, $\Re x > 0$, as

$$\mathcal{B}_x = x! \, \frac{1}{2i\pi} \int_{\mathcal{H}} e^{e^z-1} \frac{dz}{z^{x+1}}. \tag{4}$$

When generalized in this way, the Bell numbers satisfy

$$\mathcal{B}_x = \sum_{k=0}^{\infty} \left\{ {x \atop k} \right\},$$

which results from expanding the integrand of (4),

$$e^{e^z-1} = \sum_{k=0}^{\infty} \frac{(e^z-1)^k}{k!}.$$

Dobinski's formula. This classical formula [2] also generalizes. If we expand

$$e^{e^z-1} = e^{-1} \sum_{k=0}^{\infty} \frac{e^{kz}}{k!},$$

we get

$$\mathcal{B}_x = \frac{x!}{e} \sum_{k=0}^{\infty} \frac{1}{k!} \, \frac{1}{2i\pi} \int_{\mathcal{H}} e^{kz} \frac{dz}{z^{x+1}}.$$

The integral can now be evaluated by the use of Hankel's formula for the gamma function (substitute $kz = t$), and

$$\frac{1}{2i\pi} \int_{\mathcal{H}} e^{kz} \frac{dz}{z^{x+1}} = \frac{k^x}{x!};$$

hence, we have the generalized Dobinski formula,

$$\mathcal{B}_x = e^{-1} \sum_{k=0}^{\infty} \frac{k^x}{k!}. \tag{5}$$

Bernoulli numbers. Given that their exponential generating function is $z/(e^z - 1)$, it is natural to expect Bernoulli numbers to be related to Stirling numbers of type $\left\{ {x \atop -1} \right\}$. Consider first the case of an integer index n. Then

$$\left\{ {n \atop y-1} \right\} = \frac{n!}{(y-1)!} \frac{1}{2i\pi} \int_{\mathcal{H}} (e^z - 1)^{y-1} \frac{dz}{z^{n+1}}.$$

As $y \to 0$, we have by Cauchy's formula and the fact that $(y-1)! \sim \frac{1}{y}$,

$$\left\{ {n \atop y-1} \right\} \sim y \cdot n! [z^{n+1}] \frac{z}{e^z - 1} = y \frac{B_{n+1}}{n+1}.$$

This relation gives

$$\frac{d}{dy} \left\{ {n \atop y} \right\} \Bigg|_{y=-1} = \frac{B_{n+1}}{n+1} = \zeta(-n),$$

and more generally, thanks to Hankel's representation of the ζ function (see, e.g., [6]),

$$\frac{d}{dy} \left\{ {x \atop y} \right\} \Bigg|_{y=-1} = \zeta(-x).$$

In other words, Bernoulli numbers of complex index that are naturally defined by $B_{x+1} := (x+1)\zeta(-x)$ are also obtained by a simple limiting process applied to generalized Stirling numbers of index -1:

$$\frac{B_{x+1}}{x+1} = \lim_{\epsilon \to 0} \frac{1}{\epsilon} \left\{ {x \atop -1+\epsilon} \right\} = \frac{d}{dy} \left\{ {x \atop y} \right\} \Bigg|_{y=-1}.$$

Stirling cycle numbers. Formula (3) and the change of variables $z = \log(1+w)$ provide a logarithmic form of Stirling subset numbers,

$$\left\{ {x \atop y} \right\} = \frac{(x-1)!}{(y-1)!} \frac{1}{2i\pi} \int_{\mathcal{H}^\star} \left(\log \frac{1}{1+w} \right)^{-x-1} w^{y-1}\, dw, \tag{6}$$

where $\mathcal{H}^\star$ is a "raindrop contour" that is the image of $\mathcal{H}$ by $z \mapsto w = e^z - 1$. (Thus, $\mathcal{H}^\star$ starts at -1 in the lower half plane, surrounds 0 counterclockwise, and returns to -1 in the upper half plane.)

On the other hand, Cauchy's coefficient formula applied to the exponential generating function of Stirling cycle numbers gives

$$\left[{n \atop k} \right] = (-1)^{n-k} \frac{n!}{k!} \frac{1}{2i\pi} \int_{\mathcal{H}^\star} \big(\log(1+w) \big)^k \frac{dw}{w^{n+1}}. \tag{7}$$

A direct consequence of (6) and (7) is that the Stirling cycle numbers of integral arguments arise as limiting cases of generalized Stirling subset numbers as defined in (2), (3),

$$\left[{n \atop k} \right] = \lim_{\epsilon \to 0} \left\{ {-k+\epsilon \atop -n+\epsilon} \right\}.$$

One encounters once more an instance of the duality relation $\left[{x \atop y}\right] = \left\{{-y \atop -x}\right\}$ that, together with (6), confirms that Stirling numbers eventually reduce to a single family (see [3, 4]).

REFERENCES

[1] V. Adamchik, *On Stirling numbers and Euler sums*, J. Comput. Appl. Math., 79 (1997), pp. 119–130.

[2] L. Comtet, *Advanced Combinatorics*, D. Reidel, Dordrecht, The Netherlands, 1974.

[3] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics*, 2nd ed., Addison Wesley, Reading, MA, 1994.

[4] D. E. Knuth, *Two notes on notation*, Amer. Math. Monthly, 99 (1992), pp. 403–422.

[5] B. Richmond and D. Merlini, *Stirling numbers for complex arguments*, SIAM J. Discrete Math., 10 (1997), pp. 73–82.

[6] E. T. Whittaker and G. N. Watson, *A Course of Modern Analysis*, 4th ed., Cambridge University Press, London, 1927 (reprinted 1973).

SIAM J. Discrete Math.
Vol. 12, No. 2, pp. 160–207

EDGE-CONNECTIVITY AUGMENTATION WITH PARTITION CONSTRAINTS*

JØRGEN BANG-JENSEN†, HAROLD N. GABOW‡, TIBOR JORDÁN§, AND ZOLTÁN SZIGETI¶

Abstract. In the well-solved edge-connectivity augmentation problem we must find a minimum cardinality set F of edges to add to a given undirected graph to make it k-edge-connected. This paper solves the generalization where every edge of F must go between two different sets of a given partition of the vertex set. A special case of this partition-constrained problem, previously unsolved, is increasing the edge-connectivity of a bipartite graph to k while preserving bipartiteness. Based on this special case we present an application of our results in statics. Our solution to the general partition-constrained problem gives a min-max formula for $|F|$ which includes as a special case the original min-max formula of Cai and Sun [*Networks*, 19 (1989), pp. 151–172] for the problem without partition constraints.

When k is even the min-max formula for the partition-constrained problem is a natural generalization of the unconstrained version. However, this generalization fails when k is odd. We show that at most one more edge is needed when k is odd and we characterize the graphs that require such an extra edge.

We give a strongly polynomial algorithm that solves our problem in time $O(n(m+n\log n)\log n)$. Here n and m denote the number of vertices and distinct edges of the given graph, respectively. This bound is identical to the best-known time bound for the problem without partition constraints. Our algorithm is based on the splitting off technique of Lovász, like several known efficient algorithms for the unconstrained problem. However, unlike previous splitting algorithms, when k is odd our algorithm must handle obstacles that prevent all edges from being split off. Our algorithm is of interest even when specialized to the unconstrained problem, because it produces an asymptotically optimum number of distinct splits.

Key words. edge-connectivity augmentation of graphs, edge splitting, connectivity, rigidity, combinatorial algorithms

AMS subject classifications. 05C40, 05C85, 70C20, 52C25

PII. S0895480197324700

1. Introduction.

In the *edge-connectivity augmentation problem* we are given an undirected graph $G = (V, E)$ and a positive integer k; the goal is to find a smallest set F of edges on the vertex-set V for which $G' = (V, E \cup F)$ is k-edge-connected. Note that E as well as F may contain parallel edges.

This optimization problem has been extensively investigated, in part due to possible practical applications in the design of reliable networks (see [9], [21]). The first polynomial-time algorithm solving this problem was by Watanabe and Nakamura [22]. Other approaches were later developed which led to more efficient algorithms. Cai and

*Received by the editors July 18, 1997; accepted for publication (in revised form) September 22, 1998; published electronically April 29, 1999. A preliminary version of this paper appeared in *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1998, pp. 306–315.
http://www.siam.org/journals/sidma/12-2/32470.html.

†Department of Mathematics and Computer Science, Odense University, Campusvej 55, DK-5230 Odense, Denmark (jbj@imada.ou.dk).

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

§Department of Mathematics and Computer Science, Odense University, Campusvej 55, DK-5230 Odense, Denmark (jordan@cs.elte.hu). The research of this author was supported in part by Danish Natural Science Research Council grant 28808.

¶Equipe Combinatoire, Universite Paris VI, 4, place Jussieu, 75252 Paris, France (szigeti@ecp6.jussieu.fr).

Sun [3] gave a min-max characterization for the corresponding optimum value, using the splitting off method. “Splitting off” a pair su, sv of edges means replacing su and sv by a new edge uv. Using this method Frank [6] solved several extensions of the problem. For example, he showed it is tractable even if local connectivity demands or vertex costs are given. A different approach by Naor, Gusfield, and Martel [19] resulted in a faster algorithm for small values of k. The currently fastest algorithm was developed by Nagamochi and Ibaraki [17] and Nagamochi, Nakamura, and Ibaraki [18]. Their algorithm, also based on the splitting off method, runs in time $O(n(m + n \log n) \log n)$. (The parameters n and m denote the number of vertices and distinct edges of the given graph, respectively.) For further related results and the directed- and vertex-connectivity versions of the problem see the survey paper by Frank [7].

This paper solves a generalization of the above problem which we call *edge-connectivity augmentation with partition constraints*. Here a partition $\mathcal{P} = \{P_1, \ldots, P_r\}$ $(r \geq 2)$ of V is also given and the additional requirement is that each edge of the augmenting set F must connect different classes of $\mathcal{P}$. Taking $r = n$ gives the original (or in other words *unconstrained*) edge-connectivity augmentation problem as a special case. Another special case is the version where G is a bipartite graph with bipartition $V = (A, B)$ and $\mathcal{P} = \{A, B\}$. In this problem, which was open so far and was one of our main motivations, the goal is to increase the edge-connectivity of a bipartite graph while preserving bipartiteness.

The first part of our solution to the augmentation problem with partition constraints is a min-max formula for the optimum value (i.e., the minimum size of F). When k is even we show the optimum value equals the “deficiency” of a certain subpartition of V. This subpartition is either the subpartition of Cai and Sun for the unconstrained problem or a subpartition based on the partition constraints. When k is odd these subpartitions have deficiency equal to the optimum value for “most” graphs but they fail for an infinite number of graphs. The following simple example turns out to be central.

Consider the problem of augmenting a four-cycle to achieve 3-edge-connectivity. It is easy to see that the unique optimum augmentation adds two new edges, changing the graph into K_4. Now consider augmenting a four-cycle to make it 3-edge-connected and also preserve bipartiteness. It is easy to see that for this problem three new edges is optimum. But we will see that the subpartitions have deficiency only two.

Our min-max theorem shows that when k is odd any graph requires at most one more edge than the lower bounds from subpartitions. Furthermore, a graph requires an extra edge precisely when it possesses a structure generalizing the above four-cycle example or a structure generalizing the six-cycle. These structures, which we call C_4- and C_6-configurations, can exist for any odd value $k \geq 3$. A C_6-configuration is more specialized, since it exists only in graphs requiring exactly three edges to achieve k-edge-connectivity.

The second part of our solution to the partition-constrained problem is a strongly polynomial algorithm. It finds an optimum solution in the same time as the algorithm of Nagamochi, Nakamura, and Ibaraki for the unconstrained problem, $O(n(m+n \log n) \log n)$. When specialized to the unconstrained problem our algorithm is a modified version of the splitting-off algorithm of [18]. This specialization is also of interest because, like our general algorithm, it produces an asymptotically optimum number of distinct splits. This improves [17], [18] by a logarithmic factor. It gives a similar savings in space.

The splitting off method used in algorithms for edge-connectivity augmentation and a number of other problems is based on the well-known splitting off theorem of Lovász [14]. The partition-constrained problem requires a more powerful result: when k is odd we must avoid certain splits that would mistakenly force an extra edge to be added. We give a new splitting off theorem that characterizes when this can be done.

This paper also investigates an application of the problem of augmenting the connectivity of a bipartite graph (preserving bipartiteness) in the field of statics. It also shows that a variant of our problem, when the edges of the augmenting set F must lie *within* classes of a given partition, is NP-hard.

We close this section by mentioning other extensions of the edge-connectivity augmentation problem that have been investigated. Finding a k-edge-connected augmentation that has minimum cost is NP-hard (even when $k = 2$ and there are only two distinct edge costs); see [5]. Optimally augmenting a bipartite graph to achieve 2-vertex-connectivity while preserving bipartiteness was solved in [11]. Optimally augmenting to achieve k-edge-connectivity preserving simplicity of the graph is NP-hard but polynomially solvable for fixed k; see [1] and [12]. Preserving planarity was investigated in some vertex-connectivity problems; see [13]. Other types of constraints were also studied. In [4] an optimal augmenting set F is to be found which can be extended to an optimal augmentation with respect to an arbitrary higher target.

Section 2 contains definitions and some basic results. Section 3 proves the new splitting off theorem. The results on the augmentation problem are in sections 4 and 5. Our efficient algorithm is in section 6. The application to statics is in section 7, and the NP-hardness result is in section 8.

2. Terminology and some basic results. Let $G = (V, E)$ be an arbitrary undirected multigraph. A *subpartition* of V is a collection of pairwise disjoint subsets of V. The subgraph of G induced by a subset X of vertices is denoted $G[X]$. A set consisting of a single vertex v is simply denoted by v. An edge joining vertices x and y is denoted xy. Sometimes xy will refer to an arbitrary copy of the parallel edges between x and y but this will not cause any confusion. Adding or deleting an edge e from a graph G is often denoted by $G + e$ or $G - e$, respectively; adding or deleting a vertex v is similarly denoted by $G + v$ or $G - v$, respectively.

For $X, Y \subseteq V$, $d(X, Y)$ denotes the number of edges with one endvertex in $X - Y$ and the other in $Y - X$. We define the degree of a subset X as $d(X) = d(X, V - X)$. For example, $d(v)$ denotes the degree of vertex v. The degree-function of a graph G' is denoted d'. A graph $G = (V, E)$ is *k-edge-connected* if

$$d(X) \geq k \quad \text{for all} \quad \emptyset \neq X \subset V. \tag{1}$$

The degree function satisfies the following two well-known equalities.

Proposition 2.1. *Let $H = (V, E)$ be a graph. For arbitrary subsets $X, Y \subseteq V$,*

$$d(X) + d(Y) = d(X \cap Y) + d(X \cup Y) + 2d(X, Y), \tag{2}$$

$$d(X) + d(Y) = d(X - Y) + d(Y - X) + 2d(X \cap Y, V - (X \cup Y)). \tag{3}$$

The operation *splitting off* a pair of edges sv, st at a vertex s means that we replace these two edges by a new edge vt. (In the presence of parallel edges one splitting operation replaces only one copy of each of sv, st.) The notation $G_{v,t}$ denotes the graph obtained after splitting off the edges sv, st in G (the vertex s will always be clear from the context). A *complete splitting* at a vertex s (with even degree) is a sequence of $d(s)/2$ splittings of pairs of edges incident to s.

In the rest of this section let s be a specified vertex of a graph $G = (V + s, E)$ with degree function d such that $d(s)$ is even and (1) holds. We shall consider such graphs throughout the paper. Saying (1) holds in such a graph G means it holds for all $\emptyset \neq X \subset V$. (Thus (1) can hold even when $d(s) < k$.) A set $\emptyset \neq X \subset V$ is called *dangerous* if $d(X) \leq k + 1$ and *critical* if $d(X) = k$. Two sets $X, Y \subseteq V$ are *crossing* if $X - Y$, $Y - X$, $X \cap Y$, and $V - (X \cup Y)$ are all nonempty. Again these definitions refer only to subsets of V.

Edges sv, st form an *admissible pair* in G if $G_{v,t}$ still satisfies (1). It is easy to see that sv, st is not admissible if and only if some dangerous set contains both t and v.

Throughout the paper we use the following basic lemmas.

LEMMA 2.2. *If $k \geq 3$ and $d(X) \leq k+2$ for some $X \subset V$, then $G[X]$ is connected.*

Proof. Let $X = Y \cup Z$ be any partition of X into two nonempty sets. Since G satisfies (1) we get $k + 2 \geq d(X) = d(Y) + d(Z) - 2d(Y, Z) \geq k + k - 2d(Y, Z)$. Thus since $k \geq 3$ we must have that $d(Y, Z) \geq 1$ and hence the lemma follows. ∎

LEMMA 2.3. *If k is odd, then no three pairwise disjoint critical sets X, Y, Z have $N(X) \subseteq Y \cup Z$.*

Proof. Assume on the contrary that there exist such sets. Since $k = d(X) = d(X, Y) + d(X, Z)$, without loss of generality (w.l.o.g.) we may assume $d(X, Y) > k/2$. It follows that $d(X \cup Y) = d(X) + d(Y) - 2d(X, Y) < k + k - 2k/2 = k$, in contradiction to (1). ∎

LEMMA 2.4. *If k is odd, then the union of two disjoint critical sets is not critical.*

Proof. If X and Y are disjoint critical sets then $d(X \cup Y) = d(X) + d(Y) - 2d(X, Y) = 2k - 2d(X, Y)$. The right-hand side does not equal k because k is odd. ∎

LEMMA 2.5. *If k is odd, then no two critical sets are crossing.*

Proof. Suppose that X and Y are two crossing sets in V of degree k. Then $d(X \cap Y) = k$ by (2) and $d(X - Y) = k$ by (3). Thus $X - Y$ and $X \cap Y$ contradict Lemma 2.4. ∎

LEMMA 2.6. *A maximal* (*with respect to inclusion*) *dangerous set does not cross any critical set.*

Proof. Let X be a maximal dangerous set and suppose it crosses a critical set Y. The maximality of X shows $d(X \cup Y) \geq k + 2$. Thus $k + 1 + k \geq d(X) + d(Y) \geq d(X \cap Y) + d(X \cup Y) \geq k + k + 2$, a contradiction. ∎

LEMMA 2.7. *Let k be odd and let $X \subset V$ be a set of degree $k + 2$ containing two neighbors of s, x, and y, such that $d(Z) \geq k+2$ whenever $x, y \in Z \subset V$. Then X does not cross any critical set.*

Proof. The hypothesis implies sx, sy is an admissible pair. Applying Lemma 2.5 to $G_{x,y}$ gives the desired conclusion. ∎

LEMMA 2.8. *If X is dangerous, then $d(s, V - X) \geq d(s, X)$.*

Proof. By (1) we see that $k \leq d(V - X) = d(X) - d(s, X) + d(s, V - X)$. Since $d(X) \leq k+1$, this implies $d(s, V - X) \geq d(s, X) - 1$, and since $d(s)$ is even we cannot have equality. ∎

LEMMA 2.9. *If k is even, then two maximal dangerous sets X, Y which are crossing have $d(s, X \cap Y) = 0$.*

Proof. Suppose X and Y are maximal dangerous and crossing such that $d(s, X \cap Y) \geq 1$. By Proposition 2.1 and (1), we get that $d(X) = k + 1$, $d(X - Y) = k$, and $d(X \cap Y) = k$. It is easy to see that $d(X)$ is congruent to $d(X - Y) + d(X \cap Y)$ modulo 2, but this contradicts the fact that $d(X)$ is odd and each of $d(X - Y), d(X \cap Y)$ is even. ∎

LEMMA 2.10. *Any two crossing sets* $X, Y \subset V$ *satisfy* $d(X)+d(Y) \geq 2k+2d(s, X \cup Y) - d(s)$.

Proof. Observe that $d(X \cup Y) = d(V-(X \cup Y)) - d(s, V-(X \cup Y)) + d(s, X \cup Y)$ and $d(s) = d(s, X \cup Y) + d(s, V-(X \cup Y))$. Combining these and using (1) and (2) gives the lemma. ☐

LEMMA 2.11. *Let* X *be a critical set that is minimal with respect to inclusion. Then for every edge* sx, $x \in X$ *and every* $u \in X$, $G - sx + su$ *satisfies* (1).

Proof. Suppose that for some edge $e = sx$, $x \in X$ and some $u \in X$ the graph $G - e + f$, where $f = su$, does not satisfy (1). Then there must be some other critical set Y in G such that the edge sx enters Y while su does not. By the minimality of X we get that $Y - X \neq \emptyset$. Now applying (3) we get that $k + k = d(X) + d(Y) = d(X-Y) + d(Y-X) + 2d(s, X \cap Y) \geq k + k + 2$, a contradiction. ☐

Throughout the rest of this paper we always assume that $k \geq 2$ since the solutions to our problems are trivial when $k \leq 1$.

The following result of Lovász [14]—Theorem 2.12(a) below—turned out to be a useful tool in augmentation problems; see [6]. Here we formulate an easy extension, Theorem 2.12(b), which plays an important role in some of our arguments. The proof follows from the proof of part (a) given by Frank in [6, pp. 35–36]. We assume that the reader is familiar with this proof and give only the necessary additional details below.

THEOREM 2.12. *Suppose that* (1) *holds in* $G = (V+s, E)$, $k \geq 2$, *and* $d(s)$ *is even.*

(a) *For every edge* st *there exists an edge* su *such that the pair* st, su *is admissible* [14].
(b) *For every edge* st *the number of edges* su *for which the pair* st, su *is admissible is at least* $d(s)/2$ *when* k *is even and at least* $d(s)/2 - 1$ *when* k *is odd.*

Proof of (b). In the proof of (a) in [6, pp. 35–36] it is shown that for any given edge st, all the neighbors u of s for which the pair st, su is not admissible are contained in the union of at most two maximal dangerous sets, each set containing t; furthermore, when there are two such sets they are crossing.

If t is not contained in any maximal dangerous set, then st is admissible with every other edge su. If there is a unique maximal dangerous set X containing t, then st is admissible with every edge su, $u \notin X$. Lemma 2.8 shows there are at least $d(s)/2$ such edges.

The remaining case is when there are two maximal dangerous sets, say X and Y, that are crossing, with $X \cap Y$ containing t and $X \cup Y$ containing all neighbors u of s for which st, su is not admissible. Lemma 2.9 shows k is odd. This implies part (b) holds if k is even.

Continuing with the sets X and Y (and k odd), let $Z = V - (X \cup Y)$. Since G satisfies (1) we get $k \leq d(Z) = d(X \cup Y) - d(s, X \cup Y) + d(s, Z)$. (2) implies that $d(X \cup Y) = k + 2$. Hence $d(s, Z) \geq d(s, X \cup Y) - 2$. This implies at least $d(s)/2 - 1$ edges incident to s are admissible for splitting with st. ☐

Let $OPT^k(G)$ ($OPT^k_{\mathcal{P}}(G)$) denote the size of a solution to the edge-connectivity augmentation problem (with partition constraints). Hence $OPT^k(G)$ equals the minimum size of a set of new edges whose addition to G results in a k-edge-connected graph. $OPT^k_{\mathcal{P}}(G)$ is defined similarly with the additional stipulation that no new edge lies inside some $P_i \in \mathcal{P}$, where $\mathcal{P} = \{P_1, \ldots, P_r\}$ is a given partition of V.

Our algorithms are based on Frank's algorithm to calculate $OPT^k(G)$ and solve the edge-connectivity augmentation problem. Frank's algorithm uses the splitting operation and Theorem 2.12(a) as the main tool. We now summarize Frank's algorithm [6].

Frank's algorithm.
(PHASE 1) Starting with the given graph $G = (V, E)$ and the integer $k \geq 2$, add a new vertex s to V and a set F of new edges between s and some vertices of V such that

$$d'(X) \geq k \quad \text{for all} \quad \emptyset \neq X \subset V \quad \text{in} \quad G' = (V + s, E \cup F), \tag{4}$$

$$F \text{ is minimal (with respect to inclusion) subject to (4).} \tag{5}$$

(PHASE 2) If $d'(s)$ is odd in G', add a new edge sv for some arbitrary $v \in V$.
(PHASE 3) Split off admissible pairs of edges incident to s in arbitrary order, maintaining (1). When s becomes isolated, delete s.

It is clear that such an F (in the first phase) exists. It was shown in [6] that there exists a subpartition

$$\mathcal{F} = \{X_1, \ldots, X_t\} \text{ of } V \quad \text{such that} \quad |F| = \sum_1^t (k - d(X_i)), \tag{6}$$

where as usual d denotes the degree function in G. In the third phase every edge can be split off by Theorem 2.12(a). The resulting graph is an optimal k-edge-connected augmentation of G since after the second phase $OPT^k(G) \geq \frac{|F|}{2}$ by (6).

We now focus on edge-connectivity augmentation with partition constraints. An admissible pair sx, sy is called *allowed* if x and y belong to different classes of $\mathcal{P}$. A *complete allowed splitting* at a vertex s is a sequence of $d(s)/2$ splittings at s such that the ith splitting is allowed when performed in the graph obtained after performing the first $i - 1$ splittings. The following fact was also observed in [18] (in a different context).

LEMMA 2.13. *Let k be even and let $G = (V + s, E)$ be a graph satisfying* (1) *with $d(s)$ even. There exists a complete allowed splitting at vertex s if and only if $d(s, P_i) \leq d(s)/2$ for all $1 \leq i \leq r$.*

Proof. The condition $d(s, P_i) \leq d(s)/2$ is necessary for a complete allowed splitting to exist, since a pair sx, sx is not allowed.

To prove sufficiency we argue by induction on the degree of s. Choose an index j, $1 \leq j \leq r$, so that $d(s, P_j)$ is maximum. Theorem 2.12(b) implies there is an allowed pair sx, sy for any neighbor x of s belonging to P_j. Here we use the hypothesis $d(s, P_j) \leq d(s)/2$. Next we show that for $G' = G_{x,y}$, $d'(s, P_i) \leq d'(s)/2$ for all i. This is clear for $i = j$. If $i \neq j$, then $d'(s, P_i) \leq d'(s, P_j) + 1$. If equality holds, then some index $\ell \neq i, j$ has $d'(s, P_\ell) \geq 1$, since $d'(s)$ is even. This implies $d'(s, P_i) \leq d'(s)/2$. □

3. Splittings satisfying partition constraints when *k* is odd. Let $k \geq 3$ be an odd number, let $G = (V + s, E)$ be a graph such that (1) holds for G and $d(s)$ is even, and let $\mathcal{P} = \{P_1, P_2, \ldots, P_r\}$, $2 \leq r \leq |V|$ be a prescribed partition of V.

We begin by defining two types of "obstacles" that, we will show, preclude the existence of a complete allowed splitting. Let S denote the set of neighbors of s and $S_i := S \cap P_i$. Let $d_i := d(s, P_i)$.

DEFINITION 3.1. *Let $A_1 \cup A_2 \cup B_1 \cup B_2$ be a partition of V with the following properties in G for some index i, $1 \leq i \leq r$:*

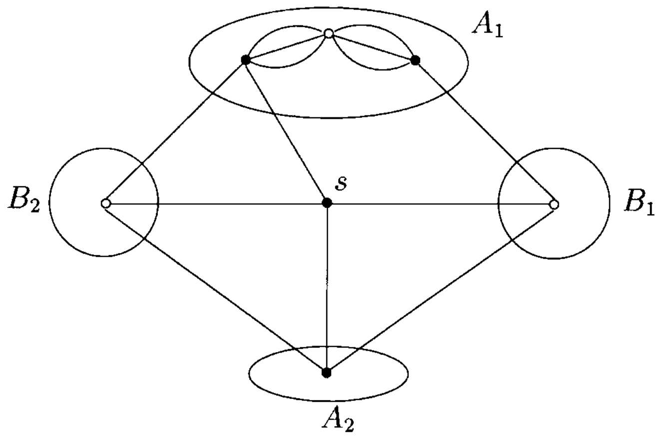


FIG. 1. *A graph $g = (V + s, E)$ with V partitioned into black and white vertices. The sets A_1, A_2, B_1, B_2 form a C_4-obstacle in G with respect to $k = 3$.*

(i) $d(X) = k$ *for* $X = A_1, A_2, B_1, B_2$;
(ii) $d(X, Y) = 0$ *for* $(X, Y) = (A_1, A_2), (B_1, B_2)$;
(iii) $S \cap X = S_i$ *for* $X = A_1 \cup A_2$ *or* $X = B_1 \cup B_2$;
(iv) $d_i = d(s)/2$.

Such a partition is called a C_4-obstacle *in* G. *The two pairs* (A_1, A_2) *and* (B_1, B_2) *listed in* (ii) *are called* nonconsecutive *while the other four pairs are called* consecutive. (*See Figure* 1.)

Note that conditions (i)–(iv) can be satisfied when k is even but a C_4-obstacle requires that k be odd. An important property of a C_4-obstacle is that s has a neighbor in each set A_1, A_2, B_1, B_2, by Lemma 2.3.

The reader will have noticed the close relation between a C_4-obstacle and the example of augmenting a four-cycle given in the introduction. The C_6-obstacle defined below has a similar motivation which we now give. Suppose we wish to augment a six-cycle to achieve 3-edge-connectivity. This can be done by adding three new edges to change the graph into $K_{3,3}$. It is easy to see that three new edges is optimum, and any optimum augmentation adds at least one new edge joining diametrically opposite vertices. Now consider augmenting a six-cycle to make it 3-edge-connected with the partition constraint imposed when $\mathcal{P}$ consists of the three sets of diametrically opposite vertices. It is easy to see that for this problem four new edges is optimum. However, we will see in section 5 that the subpartitions give a lower bound of only three.

DEFINITION 3.2. *Let* $A_1 \cup A_2 \cup B_1 \cup B_2 \cup C_1 \cup C_2$ *be a partition of* V *with the following properties in* G *for three distinct indices* a, b, c, $1 \leq a, b, c \leq r$:

(i′) $d(X) = k$ *for* $X = A_1, A_2, B_1, B_2, C_1, C_2$;
(ii′) $d(X, Y) = \frac{k-1}{2}$ *for* $(X, Y) = (A_1, B_1), (B_1, C_1), (C_1, A_2), (A_2, B_2), (B_2, C_2), (C_2, A_1)$;
(iii′) $d(X, s) = 1$ *for* $X = A_1, A_2, B_1, B_2, C_1, C_2$;
(iv′) $S \cap X = S_i$ *for* $(X, S_i) = (A_1 \cup A_2, S_a), (B_1 \cup B_2, S_b), (C_1 \cup C_2, S_c)$.

Such a partition is called a C_6*-obstacle in* G. *The six pairs listed in* (ii′) *are called* consecutive *while the other nine pairs are called* nonconsecutive. (*See Figure* 2.)

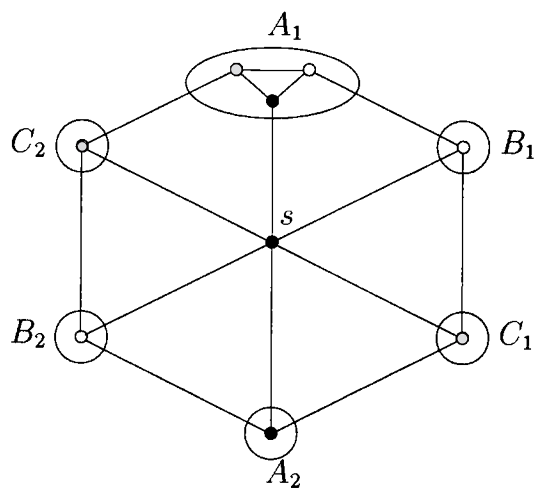


FIG. 2. *A graph $G = (V + s, E)$ with V partitioned into black, white, and shaded vertices. The sets $A_1, A_2, B_1, B_2, C_1, C_2$ form a C_6-obstacle in G with respect to $k = 3$.*

The first two lemmas allow us to split off most edges.

LEMMA 3.3. *Suppose $d(s, P_i) \leq d(s)/2$ for all $i = 1, \ldots, r$. If $d(s) \geq 6$, then for any P_i containing a neighbor of s there is an allowed pair sx, sy with $x \in P_i$.*

Proof. If $d_i < d(s)/2$, then Theorem 2.12 implies there is an allowed pair sx, sy for any neighbor x of s belonging to P_i. Hence assume $d_i = d(s)/2$. Let u be an arbitrary neighbor of s in P_i. If there is no allowed pair for the edge su, then, as mentioned in the proof of Theorem 2.12, there are (at most) two dangerous sets X and Y so that $u \in X \cap Y$ and $X \cup Y$ contains all the neighbors of s not in P_i. Since $d_i = d(s)/2$ and $d(s) \geq 6$, either $d(s, X - P_i) \geq 2$ or $d(s, Y - P_i) \geq 2$. In both cases there is a pair sy, sz of edges incident to s so that $y, z \notin P_i$ and the edges sy, sz enter a common dangerous set. (It is possible that $y = z$.) Hence sy, sz is not admissible. Theorem 2.12 shows sy is part of at least $d(s)/2 - 1$ admissible pairs. This implies that for at least one $x \in P_i$ the pair sx, sy is an allowed splitting. □

LEMMA 3.4. *Suppose $d(s, P_i) \leq d(s)/2$ for all $i = 1, \ldots, r$. If $d(s) = 4$, then there exists a complete allowed splitting at s unless G contains a C_4-obstacle. Furthermore, if this C_4-obstacle is A_1, A_2, B_1, B_2, then in graph $G - s$*

1. $d(A_1) = d(A_2) = d(B_1) = d(B_2) = k - 1$,
2. $d(A_1 \cup B_1) = d(A_1 \cup B_2) = k - 1$.

Proof. If $\max d_i = 1$, then every admissible pair is allowed and thus we are done by Theorem 2.12. Otherwise the assumption of the lemma shows $\max d_i = 2$. Assume w.l.o.g. that $d_1 = 2$. We may also assume that there are no parallel edges between the edges incident to s (since in that case we would be done again, by applying Theorem 2.12 twice). Let us denote the two neighbors of s in P_1 by x and y and denote the other two neighbors of s by u and v. Assume that neither sx, su nor sx, sv is admissible. (If either pair is admissible we are again done.) Then Lemma 2.8 implies there are two dangerous sets X and Y so that $x \in X \cap Y$, $u \in X - Y$, $v \in Y - X$, and $y \in V - (X \cup Y)$.

Now we show that $X \cap Y$, $V - (X \cup Y)$, $X - Y, Y - X$ is a C_4-obstacle. Let d

denote the degree function in G. Equation (2) implies $d(X \cup Y) = k+2$, $d(X \cap Y) = k$, and $d(X, Y) = 0$. Equation (3) (with edge sx) implies $d(X - Y) = d(Y - X) = k$ and $d(X \cap Y, V - (X \cup Y)) = 0$. Also $d(V - (X \cup Y)) = d(X \cup Y) - 2 = k$. We have now shown (i) and (ii) of Definition 3.1 and (iii) and (iv) are immediate.

In the above proof both (2) and (3) imply $d(X) = d(Y) = k + 1$. Hence the last part of the lemma follows if we set $A_1 = X \cap Y, A_2 = V - (X \cup Y), B_1 = X - Y, B_2 = Y - X$. ∎

We can now state a procedure to split off most edges in G. (The procedure is used in two proofs of section 5.) Assume that every partition class P_j has $d(s, P_j) \leq d(s)/2$ in G. Starting with G, repeatedly choose an index i such that $d(s, P_i)$ is maximum and split off an allowed pair sx, sy with $x \in P_i$. The relation $d(s, P_j) \leq d(s)/2$ for all j is maintained throughout this procedure by the argument of Lemma 2.13. Lemmas 3.3 and 3.4 show that eventually we get either a complete allowed splitting or a graph that has a C_4-obstacle and $d(s) = 4$.

LEMMA 3.5. *Suppose that the pair sx, sy is an allowed splitting and the graph $G_{x,y}$ contains a C_6-obstacle. Then G contains another allowed splitting sx, sy' (i.e., the split still involves edge sx) such that $G_{x,y'}$ contains no C_4- or C_6-obstacle.*

Proof. Suppose there is a C_6-obstacle $A_1, B_1, C_1, A_2, B_2, C_2$ in $G_{x,y}$. Denote the unique neighbors of s in each of these sets by $a_1, b_1, c_1, a_2, b_2, c_2$. By the definition of a C_6-obstacle, there exist $a \neq b \neq c \neq a$ such that $a_1, a_2 \in P_a$, $b_1, b_2 \in P_b$, $c_1, c_2 \in P_c$.

We distinguish two cases corresponding to whether x and y belong to the same set or different sets of the obstacle. In the first case we may assume w.l.o.g. that $x, y \in A_1$. In the second case x and y must belong to sets that are consecutive in the C_6-obstacle, so we may assume w.l.o.g. that $x \in A_1$ and $y \in B_1$.

Case 1. $x, y \in A_1$. At least one of c_1, b_2, say w.l.o.g. c_1, is not from the same partition class as x. Let $y' = c_1$. We claim that the pair sx, sy' is admissible in G. For suppose not and let Y be a maximal dangerous set containing x and y'. Since sx, sy is an admissible pair we have $y \notin Y$.

Lemma 2.8 shows Y contains at most four neighbors of s. Thus $Y \cup D \neq V$ for any set D of the obstacle. This implies Y and A_1 are crossing (since $y \notin Y$). It also implies Y contains any other set of the obstacle that it shares a vertex with (by Lemma 2.6). Since Y is connected (Lemma 2.2) and contains x and y', we conclude that $Y \cup A_1 = A_1 \cup B_1 \cup C_1 \cup D$ for $D = \emptyset, A_2$, or C_2. Now part (ii′) of the definition of C_6-obstacle shows that in all cases for D, $d(A_1 \cup Y) \geq \frac{k-1}{2} + \frac{k-1}{2} + 5 = k + 4$. But applying (2) shows $(k+1)+(k+2) \geq d(Y)+d(A_1) \geq d(Y \cap A_1)+d(Y \cup A_1) \geq k+(k+4)$, a contradiction. Thus Y does not exist and sx, sy' is admissible.

Suppose $G_{x,y'}$ contains an obstacle. Call this the new obstacle, as opposed to the original C_6-obstacle. The new obstacle has sets A_1', A_2', B_1', B_2' and if it is a C_6-obstacle, then it has sets C_1', C_2' as well.

Let D' be any set in the new obstacle, and let D be any set in the original obstacle except A_1 (so $d(D) = k$). In $G_{x,y'}$, neither D nor D' can contain three neighbors of s (a set in a C_4-obstacle contains ≤ 2 neighbors, if $d(s) = 6$). Hence $D \cup D' \neq V$. Furthermore, Lemma 2.5 implies the two sets do not cross. So $D \subseteq D'$ if the two sets share a vertex. (We cannot have $D' \subset D$ since that would make D crossing with another set of the new obstacle.) We shall use this principle frequently.

By renaming if necessary, we can assume that $C_1 \subset A_1'$. The inclusion is proper since C_1 has no neighbor of s in $G_{x,y'}$.

We now show $A_1' \cap A_1 \neq \emptyset$. Suppose the contrary. Then A_1' has degree k in G. By Lemma 2.2 A_1' induces a connected graph in G. Now it is easy to see (using the above principle) that we must have $A_1' = X \cup C_1 \cup Y$, where X is either B_1 or empty and Y is either empty or one of the three sets A_2, $A_2 \cup B_2$, $A_2 \cup B_2 \cup C_2$. In all cases (ii′) of Definition 3.2 implies that A_1' has degree at least $k+1$ in G. This is the desired contradiction.

We next show that $A_1 \cap A_1'$ contains no neighbor of s in $G_{x,y'}$. Since A_1' induces a connected graph in G (Lemma 2.2) and has at most two neighbors of s in $G_{x,y'}$, our principle implies that $B_1 \subset A_1'$ and that $B_2 \cap A_1' = \emptyset$. Applying (2) to the sets A_1 and A_1' (of degree $k+2$ and k, respectively, in $G_{x,y'}$) we see that in $G_{x,y'}$ $A_1' \cup A_1$ has degree at most $k+2$. Now it is easy to prove, by examining the neighbors of s in $G_{x,y'}$, that $A_1' \cap C_2 = A_1' \cap A_2 = \emptyset$. Hence $A_1' \cup A_1 = A_1 \cup B_1 \cup C_1$. Thus in $G_{x,y'}$ we have $d(A_1' - A_1) = k+1$. Then by (3), $d(A_1 \cap A_1', V - (A_1 \cup A_1')) = 0$ holds in $G_{x,y'}$. This implies the desired conclusion.

Suppose first that the new obstacle is a C_4-obstacle. This implies two of the sets A_2', B_1', B_2' contain exactly two neighbors of s. Lemma 2.7 shows these sets do not cross A_1. Now our principle shows at least one of these sets is the union of exactly two sets of the original obstacle. But this contradicts Lemma 2.4.

Now suppose that the new obstacle is a C_6-obstacle. Since the unique neighbor b_1 of s in A_1' in the graph $G_{x,y'}$ belongs to P_b, it follows from Definition 3.2 and our principle that $A_2' = B_2$. Thus $G_{x,y'} - s$ has a path from A_1' to A_2' that contains vertices in only one other new obstacle set (namely the new obstacle set containing A_2). This contradicts the fact that in a C_6-obstacle any path avoiding s from A_1' to A_2' goes through at least two other obstacle sets.

Case 2. $x \in A_1$ and $y \in B_1$. As in Case 1 at least one of c_1, b_2 is not from the same partition class as x. In the first case we let $y' = c_1$ and in the second case we let $y' = b_2$. We claim that the pair sx, sy' is admissible in G. Suppose not and let Y be a maximal dangerous set containing x, y' in G. Since $d(A_1) = k$, it follows from Lemma 2.6 and Lemma 2.8 that $A_1 \subset Y$. Suppose first that $y' = c_1$. Since Y is connected by Lemma 2.2, we see as we did in Case 1 that either $B_1 \subset Y$ or $A_2 \cup B_2 \cup C_2 \subset Y$, both of which contradict Lemma 2.8. If $y' = b_2$, then the only case which does not lead to a contradiction as above is $Y = B_2 \cup C_2 \cup A_1$. But then $d(Y) = k+2$, a contradiction. Hence the pair sx, sy' is admissible for splitting.

Suppose that $G_{x,y'}$ contains a new obstacle. Assume first that $y' = c_1$. As in Case 1 we let A_1' be that set of the new obstacle such that $C_1 \subset A_1'$. The same argument as in Case 1 shows that $A_1' \cap A_1 \neq \emptyset$. Lemma 2.5 (applied in $G_{x,y'}$) shows that $A_1 \subset A_1'$. Now as in Case 1 we can prove that $B_1 \subset A_1'$. But then A_1' contains three neighbors of s in $G_{x,y'}$, a contradiction.

Assume finally that $y' = b_2$. Let A_1' be that set of the new obstacle that contains B_2. Again we deduce that $A_1 \subset A_1'$. Using the fact that A_1' induces a connected set in G and has at most two neighbors of s in $G_{x,y'}$, we get $A_1' = B_2 \cup C_2 \cup A_1$. Thus A_1' has precisely two neighbors of s. Thus the new obstacle is a C_4-obstacle A_1', B_1', A_2', B_2'. In graph $G_{x,y'}$, B_1 contains two neighbors of s and has degree k. Hence Lemma 2.5 and Definition 3.1 show that w.l.o.g. $B_1' = B_1$. Now it is easy to see that each of the sets $A_1' \cup A_2'$ and $B_1' \cup B_2'$ contains neighbors of s from two different sets of the partition $\mathcal{P}$, contradicting Definition 3.1. ∎

THEOREM 3.6. *There exists a complete allowed splitting at vertex s in G if and only if*

(a) $d_i \leq d(s)/2$ *for all* $1 \leq i \leq r$,

(b) G *contains no* C_4*- or* C_6*-obstacle.*

Proof (necessity). Suppose that there exists a complete allowed splitting off at vertex s in G. Since all the splittings are allowed, (a) is satisfied.

Suppose that G contains a C_4-obstacle. By (iii) and (iv) of Definition 3.1 the same partition remains a C_4-obstacle each time a split is executed. But then after executing all the splits the sets A_1, B_1, and B_2 contradict Lemma 2.3.

Now suppose that G contains a C_6-obstacle. Assume that a_1 is the unique neighbor of s in A_1 and suppose that the edge sa_1 was split off with sx. Then x cannot lie in A_2 because the splitting is allowed. Moreover, it can be neither in B_1 nor in C_2 since $A_1 \cup B_1$ and $A_1 \cup C_2$ are dangerous sets. Thus x is in either C_1 or B_2, say in C_1. Let us split off the pair sa_1, sx. Then $A_1 \cup B_1 \cup C_1, A_2, B_2, C_2$ is a C_4-obstacle, a contradiction by the above argument. Hence, if G contains a C_6-obstacle, it cannot have a complete allowed splitting at s. ☐

Proof (sufficiency). Assume that G satisfies (a) and (b). We use induction on $d(s)$ to show that there exists a complete allowed splitting at vertex s in G. If $d(s) = 2$, then the result clearly holds. If $d(s) = 4$, then the result follows from Lemma 3.4. Hence assume that $d(s) \geq 6$.

Without loss of generality assume d_1 is the maximum value d_i. Using Lemma 3.3, split off an allowed pair sx, sy, where $x \in P_1$. $G_{x,y}$ satisfies condition (a) of Theorem 3.6 (as in the proof of Lemma 2.13). If $G_{x,y}$ contains no obstacle, then by induction we are done.

Suppose $G_{x,y}$ contains a C_6-obstacle. It follows from Lemma 3.5 that we can replace sx, sy by another splitting sx, sy' such that $G_{x,y'}$ has no C_4-obstacle and no C_6-obstacle. Thus we are done by induction.

The remaining case is when $G_{x,y}$ contains a C_4-obstacle, say A_1, A_2, B_1, B_2. Then x and y are either in the same set or in two consecutive sets of the obstacle. Hence we may assume w.l.o.g. that either $x, y \in A_1$ (Case 1) or $x \in A_1$ and $y \in B_1$ (Case 2). In both cases we shall first show the following is true (after possibly renaming the sets A_1, A_2, B_1, B_2): There is another allowed split sx', sy' that shares an edge, say sz, with sx, sy and has $x' \in P_j$, where $d_j = d_1$. Furthermore, $z \in A_1$, and choosing vertex w' so $\{x', y'\} = \{w', z\}$, $w' \in A_2$. Also, choose vertex w so $\{x, y\} = \{w, z\}$.

In Cases 1 and 2 below fix the graph as G.

Case 1. $x, y \in A_1$. This means that in G, $d(A_1) = k + 2$, $d(A_2) = d(B_1) = d(B_2) = k$.

If A_2 contains a neighbor u of s not in P_1, then let $x' := x$ and $y' := u$. Otherwise, let x' be any neighbor of s in A_2 (x' exists by Lemma 2.3) and let $y' := y$. Note that either $x = x'$ or $y = y'$, and the two pairs sx, sy and sx', sy' are as desired.

CLAIM 3.7. *sx', sy' is admissible.*

Proof. If sx', sy' is not admissible, then there exists a maximal dangerous set Y containing x' and y'. Y does not contain vertex w because sx, sy is admissible. Consequently $A_1 - Y \neq \emptyset$. $G[Y]$ is connected by Lemma 2.2. This implies that Y intersects $B_1 \cup B_2$ (say B_1) because there is no edge between A_1 and A_2. Lemma 2.6 shows that Y contains both A_2 and B_1.

We show that $Y \cap B_2 = \emptyset$. Otherwise Lemma 2.6 shows $B_2 \subseteq Y$. Then $d(s, Y) \geq \frac{d(s)-2}{2} + 1 + 1$, because $d(s, B_1 \cup B_2) = \frac{d(s)-2}{2}, d(s, A_2) \geq 1, d(s, A_1 \cap Y) \geq 1$. This is a contradiction against Lemma 2.8.

Note that $d(s, A_1 \cup Y) \geq \frac{d(s)-2}{2} + 2 + 1 = \frac{d(s)}{2} + 2$. Now Lemma 2.10 gives $(k + 2) + (k + 1) \geq d(A_1) + d(Y) \geq 2k + 2d(s, A_1 \cup Y) - d(s) \geq 2k + 4$. This contradiction shows sx', sy' is admissible. ☐

Case 2. $x \in A_1$ and $y \in B_1$. This means that in G, $d(A_1) = d(A_2) = d(B_1) = d(B_2) = k$. If A_2 contains a neighbor u of s not in P_1, then let $x' := x$ and $y' := u$.

Otherwise, A_1 must contain a neighbor of s not in P_1 because this partition is not a C_4-obstacle in G. Thus in the graph $G_{x,y}$ all the neighbors of s in $B_1 \cup B_2$ belong to the same class P_j, $j \neq 1$. This implies that in G, $d_j \geq \frac{d(s)-2}{2}$ and $d_1 \leq \frac{d(s)-2}{2} - 1 + 1$. Since d_1 was maximum, it follows that $d_1 = d_j = \frac{d(s)-2}{2}$ in G and $y \notin P_j$. In this case let x' be any neighbor of s in B_2 (and thus $x' \in P_j$) and $y' := y$. Note that although $x' \notin P_1$ it belongs to a class P_j, where d_j is also maximum. Now rename the sets A_1, A_2, B_1, B_2 as B_1, B_2, A_2, A_1, respectively. In both cases the two pairs sx, sy and sx', sy' are as desired.

CLAIM 3.8. *sx', sy' is admissible.*

Proof. If sx', sy' is not admissible, then there exists a maximal dangerous set Y containing x' and y'. The vertex w does not belong to Y because the pair sx, sy was admissible. Thus $B_1 - Y \neq \emptyset$. Hence Lemma 2.6 shows $A_1 \cup A_2 \subseteq Y$.

$G[Y]$ is connected by Lemma 2.2. This implies that Y intersects $B_1 \cup B_2$ because there is no edge between A_1 and A_2. If Y intersects B_i, then Lemma 2.6 shows Y contains B_i. Hence $i = 2$ and $B_2 \subseteq Y$. Then $d(s, Y) \geq d(s)/2 + 1$ because $d(s, A_1 \cup A_2) = d(s)/2$ and $d(s, B_2) \geq 1$ by Lemma 2.3. This contradicts Lemma 2.8. □

We have now found x', y' as desired. As shown previously, if $G_{x',y'}$ contains no obstacle or contains a C_6-obstacle, then we are done. The remaining possibility is when $G_{x',y'}$ contains a C_4-obstacle. This cannot occur if the hypotheses of the theorem hold, because of the following lemma. (The lemma will also be used for our algorithm.) Let a C_4-obstacle in $G_{x',y'}$ be A_1^*, A_2^*, B_1^*, B_2^*, where w.l.o.g. A_1^* contains vertex z of the common edge sz of the two splittings.

LEMMA 3.9. *After possibly interchanging B_1 and B_2, and/or B_1^* and B_2^*, the sets $A_1 \cap A_1^*$, B_1^*, A_2^*, B_2, A_2, B_1 form a C_6-obstacle in G.*

Proof.

CLAIM 3.10. *No two of the eight given obstacle sets cover V.*

Proof. Let C (C^*) be any set from the four sets A_1, A_2, B_1, B_2 ($A_1^*, A_2^*, B_1^*, B_2^*$). Assume $C \cup C^* = V$. Then C (C^*) contains the other three sets from the second (first) obstacle. By Lemma 2.3, each of the eight obstacle sets contains a neighbor of s in G. Using this and property (iv) for both C_4-obstacles we get that $d(s, C - C^*) \geq \frac{d(s)-2}{2} + 1 = \frac{d(s)}{2}$, and equality may hold only if $x', y' \in C^*$. Similarly, $d(s, C^* - C) \geq d(s)/2$ and equality may hold only if $x, y \in C$. However the two splits share an edge, so if equality holds in both relations, then $d(s, C \cap C^*) > 0$. Thus $d(s) > d(s, C - C^*) + d(s, C^* - C) \geq d(s)/2 + d(s)/2$, a contradiction. □

Case A. $d(A_1) = d(A_1^*) = k + 2$. We show that the sets of Lemma 3.9 have the properties of a C_6-obstacle using the following nine claims. Note that in Case A, $z \in A_1 \cap A_1^*$ and $w' \in A_2 \cap A_1^*$.

CLAIM 3.11. *A_1 and A_1^* are crossing sets.*

Proof. As just mentioned $A_1^* \cap A_1$ contains z so it is nonempty. $A_1 \cup A_1^* \neq V$ by Claim 3.10. $A_1^* \not\subseteq A_1$ because of vertex w'. Furthermore, $A_2 \subseteq A_1^*$. (This follows from Lemma 2.7 since $w' \in A_1^* \cap A_2$ and $z \in A_1^* - A_2$.) Since $d(s, A_1 \cup A_2) = \frac{d(s)-2}{2} + 2 = d(s, A_1^*) + d(s, A_2^*)$, we cannot also have $A_1 \subseteq A_1^*$ (since $d(s, A_2^*) > 0$). Hence A_1 and A_1^* are indeed crossing sets. □

By Lemma 2.2, $G[A_1^*]$ is connected. Thus A_1^* intersects $B_1 \cup B_2$. We also have that A_1 intersects $B_1^* \cup B_2^*$. To prove this note that Claim 3.11 shows $A_1 \cap (A_2^* \cup B_1^* \cup B_2^*) \neq \emptyset$. If $A_1 \cap A_2^* \neq \emptyset$, the desired conclusion follows by applying Lemma 2.2 as above. Now w.l.o.g. assume that A_1 intersects B_1^* and A_1^* intersects B_1.

CLAIM 3.12. *$B_1 \cup A_2 \subseteq A_1^*$, $B_1^* \cup A_2^* \subseteq A_1$, and $B_2 = B_2^*$.*

Proof. Since $w' \in A_2 \cap A_1^*$, Lemma 2.7 shows $B_1 \cup A_2 \subseteq A_1^*$. A_1^* cannot intersect B_2. (If it did, Lemma 2.7 implies $B_2 \subseteq A_1^*$. Then $A_1 \cup A_1^* = V$, contradicting Claim 3.10.) Hence $A_1^* - A_1 = B_1 \cup A_2$.

We now show that $A_1 - A_1^* = B_1^* \cup A_2^*$ (as claimed) by eliminating all other possiblities. Lemma 2.7 shows that A_1 does not cross any of the sets A_2^*, B_1^*, B_2^*. Recall that B_1^* intersects A_1, so $B_1^* \subseteq A_1 - A_1^*$. Suppose $A_1 - A_1^* = B_1^*$. Then $B_2 = A_2^* \cup B_2^*$, contradicting Lemma 2.4. Suppose $A_1 - A_1^* = B_1^* \cup B_2^*$. Then Lemma 2.2 shows $d(A_1 - A_1^*) \geq k+3$. But then $d(s, A_1 \cap A_1^*) \geq 1$ and (3) give $k+2+k+2 = d(A_1) + d(A_1^*) = d(A_1 - A_1^*) + d(A_1^* - A_1) + 2d(s, A_1 \cap A_1^*) \geq k+3+k+2$, a contradiction. Claim 3.10 shows we cannot have $A_2^* \cup B_1^* \cup B_2^* \subset A_1$. We conclude that $A_1 - A_1^* = B_1^* \cup A_2^*$.

The third part of the claim follows from the other two. □

Note that Claim 3.12 suffices for Lemma 3.9 when there are only two partition classes, because it shows the lemma is vacuous: (B_1, B_2) is a nonconsecutive pair in A_1, A_2, B_1, B_2 so all neighbors of s in B_1 and B_2 are in the same partition class. But (A_1^*, B_2^*) is a consecutive pair in the other obstacle, so all neighbors of $B_1 \subseteq A_1^*$ and $B_2 \subseteq B_2^*$ are in different partition classes. We continue the analysis for the general case.

CLAIM 3.13. *$d(A_1 - A_1^*) = k+1 = d(A_1^* - A_1)$, $d(s, A_1^* \cap A_1) = 1$, and $d(B_1, A_2) = \frac{k-1}{2} = d(B_1^*, A_2^*)$.*

Proof. By Lemma 2.4, $d(A_1 - A_1^*)$ and $d(A_1^* - A_1)$ are at least $k+1$, since they are both the disjoint union of two sets of degree k. Then (3) and the fact that $z \in A_1^*$ give $k+2+k+2 = d(A_1) + d(A_1^*) = d(A_1 - A_1^*) + d(A_1^* - A_1) + 2d(A_1^* \cap A_1, V+s-(A_1^* \cup A_1)) \geq k+1+k+1+2$. Thus equality holds everywhere, implying that $d(A_1 - A_1^*) = k+1 = d(A_1^* - A_1)$ and $d(A_1^* \cap A_1, V+s-(A_1^* \cup A_1)) = 1$. In other words, $d(s, A_1^* \cap A_1) = 1$ and $d(A_1^* \cap A_1, B_2) = 0$.

The last relation, $d(B_1, A_2) = \frac{k-1}{2}$, follows from $k+1 = d(A_1^* - A_1) = d(B_1 \cup A_2) = d(B_1) + d(A_2) - 2d(B_1, A_2) = k + k - 2d(B_1, A_2)$. Similarly $d(B_1^*, A_2^*) = \frac{k-1}{2}$. □

CLAIM 3.14. *$d(A_1^* \cup A_1) \geq k+4$.*

Proof. Obviously $d(s, A_1 \cup A_2) = d(s, B_1 \cup B_2) + 2$. Thus by Lemma 2.3, $d(s, A_1^* \cup A_1) = d(s, A_1 \cup A_2 \cup B_1) \geq d(s, A_1 \cup A_2) + 1 = d(s, B_1 \cup B_2) + 3 \geq d(s, B_2) + 4$. So $k = d(B_2) = d(A_1^* \cup A_1) - d(s, A_1^* \cup A_1) + d(s, B_2) \leq d(A_1^* \cup A_1) - 4$, as desired. □

CLAIM 3.15. *$d(A_1 \cap A_1^*) = k$ and $d(A_1, A_1^*) = 0$.*

Proof. Applying (2) for A_1 and A_1^* and using Claim 3.14 we have $k+2+k+2 = d(A_1) + d(A_1^*) = d(A_1 \cap A_1^*) + d(A_1 \cup A_1^*) + 2d(A_1, A_1^*) \geq k + (k+4)$. Thus equality holds everywhere and the claim follows. □

CLAIM 3.16. *$d((A_1 \cap A_1^*) \cup B_1^*) = k+1 = d((A_1 \cap A_1^*) \cup B_1)$.*

Proof. Lemma 2.4 implies $d((A_1 \cap A_1^*) \cup B_1^*) \geq k+1$. Thus using Claim 3.13, $k+2 = d(A_1) = d((A_1 \cap A_1^*) \cup B_1^*) + d(A_2^*) - 2d(A_2^*, B_1^* \cup (A_1 \cap A_1^*)) \geq k + 1 + k - 2\frac{k-1}{2} = k+2$. This gives the first part of the claim. The second part is similar. □

CLAIM 3.17. *$d(A_1 \cap A_1^*, B_1^*) = \frac{k-1}{2} = d(A_1 \cap A_1^*, B_1)$ and $d(s, B_1^*) = 1 = d(s, B_1)$.*

Proof. We prove both equalities for B_1^*. $k + k - 2d(A_1 \cap A_1^*, B_1^*) = d(A_1 \cap A_1^*) + d(B_1^*) - 2d(A_1 \cap A_1^*, B_1^*) = d((A_1 \cap A_1^*) \cup B_1^*) = k+1$ by Claim 3.16. The first part of the claim follows.

Now using Claim 3.13, $1 \leq d(s, B_1^*) \leq d(B_1^*) - d(A_1 \cap A_1^*, B_1^*) - d(A_2^*, B_1^*) = k - \frac{k-1}{2} - \frac{k-1}{2} = 1$. Thus $d(s, B_1^*) = 1$. □

By Claim 3.16 $d((A_1 \cap A_1^*) \cup B_1^*) = k+1$ so this set does not contain both x and y. It follows that $w \in A_2^*$. (Recall also $w' \in A_2$.)

Let the unique neighbor of s in B_1^* (B_1) belong to P_q (P_r).

CLAIM 3.18. $d(s, A_2) = 1 = d(s, A_2^*) = 1$.

Proof. After splitting off sx', sy', $B_1 \subseteq A_1^*$ and $S_r \cap B_1 \neq \emptyset$. Thus Definition 3.1(iii) shows $S_r \cap B_2^* = \emptyset$. Similarly, $S_q \cap B_2 = \emptyset$. By Lemma 2.3, $B_2 = B_2^*$ contains at least one neighbor of s; thus $B_1 \cup B_2$ intersects at least two classes of $\mathcal{P}$. This implies that after splitting off sx, sy in G, $S_q = (A_1 \cup A_2) \cap S$ and $d_q = \frac{d(s)-2}{2}$. However, considering again $G_{x',y'}$, A_1^* cannot intersect S_q since $S_q \cap B_1^* \neq \emptyset$. Thus $d(s, A_2) = 1$. Similarly, $S \cap (A_1^* \cup A_2^*) = S_r$ and $d(s, A_2^*) = 1$. □

CLAIM 3.19. $d(A_2, B_2) = \frac{k-1}{2} = d(A_2^*, B_2)$ *and* $d(s, B_2) = 1$.

Proof. Since $d(A_2, s) = 1, d(A_2, B_1) = \frac{k-1}{2}$ and $B_1 \cup B_2 \cup s$ contains all the neighbors of A_2, we see that $d(A_2, B_2) = \frac{k-1}{2}$. Similarly, $d(A_2^*, B_2) = \frac{k-1}{2}$. Then from $d(B_2) = k$ we obtain that $d(s, B_2) = 1$. □

To finish showing we have a C_6-obstacle we must prove (iv$'$) holds. It follows from the claims that $d(s) = 6$. In the proof of Claim 3.18, it was shown that $d_q = d_r = \frac{d(s)-2}{2} = 2$ and $S \cap (A_2 \cup B_1^*) = S_q, S \cap (A_2^* \cup B_1) = S_r$. It was also shown that $B_2 \cap (S_q \cup S_r) = \emptyset$. Moreover, since both splittings are allowed we have $(A_1 \cap A_1^*) \cap (S_q \cup S_r) = \emptyset$. It remains to show that if the unique neighbor of s in B_2 (in $A_1 \cap A_1^*$) belongs to P_d (P_e) of the partition $\mathcal{P}$, then $d = e$.

Note that $x \in P_r \cup P_e$. Hence A_2 contains a neighbor of s not in the same partition class as x. Hence Case 1 chooses $x' = x = z$. Since $x \in P_1$, $d_e = \max d_j = 2$. Hence $d = e$ and we are done.

Case B. At least one of $d(A_1)$ and $d(A_1^*)$ is k.

CLAIM 3.20. *The partitions of the two obstacles coincide.*

Proof. All four sets of one of the obstacles have degree k in G. Claim 3.10 and Lemmas 2.4 and 2.5 show that any set of degree k in the other obstacle equals one of the sets in the first obstacle. Since there are at least three such sets, the fourth set of both obstacles coincide, too. □

The claim implies in particular that $d(A_1) = d(A_1^*) = k$. Recalling Case 2 we have $x \in A_1, y \in B_1$, and $w' \in A_2$. Now it is easy to see that

($*$) (A_1, A_2) and (B_1, B_2) are both (non)consecutive pairs in the second (first) C_4-obstacle.

In $G_{x,y}$ either $A_1 \cup A_2$ or $B_1 \cup B_2$ contains neighbors of s from only one class P_j of $\mathcal{P}$, by ($*$) and Definition 3.1(iii). In $G_{x',y'}$ both sets of this union contain neighbors of s in P_j. (If the union is $A_1 \cup A_2$, Lemma 2.3 shows each obstacle set in $G_{x',y'}$ contains a neighbor of s.) This contradicts Definition 3.1(iii) by ($*$). We conclude that Case B cannot occur. This completes the proof of both Lemma 3.9 and Theorem 3.6. □

4. Increasing the edge-connectivity to an even number with partition constraints. Let $H = (V, E)$ be a graph with a partition $\mathcal{P} = \{P_1 \cup \ldots \cup P_r\}$, $r \geq 2$ of V. For the rest of this paper fix Φ to be the maximum of the following $r+1$ quantities:

$$\alpha = \max \left\{ \left\lceil \frac{\sum_{X \in \mathcal{F}} (k - d(X))}{2} \right\rceil : \mathcal{F} \text{ a subpartition of } V \right\};$$

$$\beta_i = \max \left\{ \sum_{Y \in \mathcal{F}} (k - d(Y)) : \mathcal{F} \text{ a subpartition of } P_i \right\} \quad \text{for} \quad i = 1, \ldots, r.$$

Observe that for any k of arbitrary parity, $OPT_{\mathcal{P}}^k(H) \geq \Phi$. In proof, clearly $OPT_{\mathcal{P}}^k(H) \geq OPT^k(H) = \alpha$. Also, $OPT_{\mathcal{P}}^k(H) \geq \beta_i$ since we are not allowed to add an edge inside any P_i.

LEMMA 4.1. *If k is even, then $OPT^k_{\mathcal{P}}(H) = \Phi$.*

Proof. We give a constructive proof to show $OPT^k_{\mathcal{P}}(H) \leq \Phi$. Steps 1 and 2 of the proof are valid even when k is odd and will be used in the next section.

Step 1. Execute the first two phases of Frank's algorithm (described in section 2) with input H, k. Now there exists a subpartition $\mathcal{F}$ of V such that $d(s)/2 = OPT^k(H) = \lceil \frac{\sum_{X \in \mathcal{F}}(k-d(X))}{2} \rceil$. In particular $d(s)/2 \leq \alpha$ (actually, equality holds).

Step 2. If there is some P_i for which $d(s, P_i) > d(s)/2$, then repeat the following three steps to make $d(s, P_i) = d(s)/2$.

Step 2(a). For each edge su, $u \in P_i$ let X_u be a minimal critical set containing u.

Step 2(b). If $X_u \not\subseteq P_i$ for some $u \in P_i$, then choose any $v \in X_u - P_i$ and replace the edge su by an edge sv. Lemma 2.11 shows that the resulting graph still satisfies (1). If we now have $d(s, P_i) = d(s)/2$, then Step 2 is complete, so continue to Step 3. Otherwise (we still have $d(s, P_i) > d(s)/2$) go back to Step 2(a).

Step 2(c). At this point $X_u \subseteq P_i$ for every edge su, $u \in P_i$. Let $\mathcal{F}_i$ be the collection of maximal sets X_u. The same proof as [6, p. 35] shows that $\mathcal{F}_i$ is a subpartition of P_i and $d(s, P_i) = \sum\{k - d(Y, V - Y) : Y \in \mathcal{F}_i\}$. Now add $d(s, P_i) - d(s, V - P_i)$ edges from s to $V - P_i$ arbitrarily. This makes $d(s)/2 \leq \beta_i$.

Step 3. In the current graph G, (1) holds and $d(s, P_j) \leq d(s)/2$ for $j = 1, \ldots, r$. It follows from Lemma 2.13 that we can obtain a complete allowed splitting off at s. This proves $OPT^k_{\mathcal{P}}(H) \leq d(s)/2 \leq \Phi$ as desired. □

We close this section by noting that even when k is odd, Steps 1 and 2 make $\Phi = d(s)/2 \geq d(s, P_i)$ for all $i = 1, \ldots, r$. In proof, the argument above shows $d(s)/2 \leq \Phi$ for k odd. Step 1 sets $d(s)/2 = \alpha$ so the final graph has $d(s)/2 \geq \alpha$. The final graph has $d(s)/2 \geq d(s, P_i)$ for all i by Step 2. Also $d(s, P_i) \geq \beta_i$ since (1) holds. The desired relations follow.

5. Increasing the edge-connectivity to an odd number with partition constraints. This section treats the more complicated case when the target connectivity is an odd number. Let $H = (V, E)$ be a fixed graph, $k \geq 3$ be an odd integer and $\mathcal{P} = \{P_1, \ldots, P_r\}$ be a prescribed partition of V. Furthermore, let $G = (V+s, E \cup F)$ be the graph obtained by executing Steps 1 and 2 of the algorithmic proof of Lemma 4.1.

We begin with a simple upper bound for partition-constrained augmentation. The proof generalizes the fact that the four-cycle can be augmented to a 3-edge-connected bipartite graph by duplicating three edges.

LEMMA 5.1. $OPT^k_{\mathcal{P}}(H) \leq \Phi + 1$.

Proof. The remark at the end of section 4 shows that every partition class P_i has $d(s, P_i) \leq d(s)/2$ in G. Now execute the procedure given after Lemma 3.4 to split off edges of G. As remarked, eventually either (i) we get a complete allowed splitting, or (ii) we get a C_4-obstacle in the (modified) graph G having $d(s) = 4$.

The proof of Lemma 4.1 and its discussion shows $\Phi \leq OPT^k_{\mathcal{P}}(H)$ and the original graph G has $d(s)/2 = \Phi$. Thus in case (i) we have $OPT^k_{\mathcal{P}}(H) = \Phi$. In case (ii) we get a sequence of $\Phi - 2$ allowed splittings that add a set of $\Phi - 2$ edges F' to G. Thus it suffices to prove that we can make the graph $H + F'$ k-edge-connected by adding three edges without violating the partition constraints.

In the rest of this argument fix G as the final graph having $d(s) = 4$. Lemma 3.4 shows G contains a C_4 obstacle A_1, A_2, B_1, B_2 such that each of the sets A_1, A_2, B_1, B_2, $A_1 \cup B_1, A_1 \cup B_2$ have degree $k - 1$ in $G - s$. Let us denote the unique neighbor of s in each of the four sets A_1, A_2, B_1, B_2 by a_1, a_2, b_1, b_2, respectively. Since Frank's algorithm makes $G - s$ k-edge-connected by splitting off two edges of G, it must add the edges a_1a_2 and b_1b_2.

We show that $G-s$ is $(k-1)$-edge-connected. Suppose some $Z \subset V$ has degree at most $k-2$ in $G-s$. We can assume $a_1 \in Z$ (if not, replace Z by its complement). Then $a_2 \notin Z$ and Z contains precisely one of b_1, b_2—say, b_2. Thus Z forms a crossing pair with $A_1 \cup B_1$. Note that b_1b_2 does not leave either of the sets $Z \cap (A_1 \cup B_1)$, $Z \cup A_1 \cup B_1$. Hence (2) shows $(k-2)+(k-1) \geq d(Z)+d(A_1 \cup B_1) \geq d(Z \cap (A_1 \cup B_1)) + d(Z \cup A_1 \cup B_1) \geq (k-1)+(k-1)$, a contradiction. We conclude that $G-s$ is $(k-1)$-edge-connected.

A degree $k-1$ set of $G-s$ contains exactly one of the vertices a_1, a_2 or exactly one of the vertices b_1, b_2 (by Frank's algorithm). Now it is easy to see that adding the edges a_1b_1, b_1a_2, a_2b_2 makes $G-s$ k-edge-connected. □

It is possible that our graph $G=(V+s, E \cup F)$ contains a C_4- or C_6-obstacle and yet $OPT^k_{\mathcal{P}}(H)=\Phi$. This can occur when we can move certain edges incident to s so that a complete allowed splitting exists. This motivates our next two definitions, which identify the configurations in the given graph H that force $OPT^k_{\mathcal{P}}(H) > \Phi$. (This is proved in Theorem 5.8.)

DEFINITION 5.2. *Let X_1, X_2, Y_1, Y_2 be a partition of V with the following properties in H:*

(i) $d(A) < k$ *for* $A = X_1, X_2, Y_1, Y_2$.

(ii) $d(A,B) = 0$ *for* $(A,B) = (X_1, X_2), (Y_1, Y_2)$.

(iii) *There exist subpartitions $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}'_1, \mathcal{F}'_2$ of X_1, X_2, Y_1, Y_2, respectively, such that for A ranging over X_1, X_2, Y_1, Y_2, and $\mathcal{F}$ the corresponding subpartition of A, $k-d(A)=\sum_{U\in\mathcal{F}}(k-d(U))$. Furthermore for some $i \leq r$, P_i contains every set of either $\mathcal{F}_1 \cup \mathcal{F}_2$ or ${\mathcal{F}'}_1 \cup {\mathcal{F}'}_2$.*

(iv) $(k-d(X_1))+(k-d(X_2)) = (k-d(Y_1))+(k-d(Y_2)) = \Phi$.

Such a partition is called a C_4-configuration *of H.* (*See* Figure 3.)

As with C_4-obstacles, k must be odd in a C_4-configuration.

DEFINITION 5.3. *Let $X_1 \cup X_2 \cup Y_1 \cup Y_2 \cup Z_1 \cup Z_2$ be a partition of V with the following properties in H:*

(i′) $d(A) = k-1$ *for* $A = X_1, X_2, Y_1, Y_2, Z_1, Z_2$.

(ii′) $d(A,B) = \frac{k-1}{2}$ *for* $(A,B) = (X_1,Y_1), (Y_1,Z_1), (Z_1,X_2), (X_2,Y_2), (Y_2,Z_2), (Z_2,X_1)$.

(iii′) *For three distinct partition classes P_a, P_b, P_c there are six sets of degree $k-1$, $X'_1, X'_2, Y'_1, Y'_2, Z'_1, Z'_2$ contained in $X_1 \cap P_a, X_2 \cap P_a, Y_1 \cap P_b, Y_2 \cap P_b, Z_1 \cap P_c, Z_2 \cap P_c$, respectively.*

(iv′) $\Phi = 3$.

Such a partition is called a C_6-configuration *of H.*

LEMMA 5.4. *If H contains a C_4-configuration, then it cannot be made k-edge-connected by adding Φ edges without violating the partition constraints.*

Proof. Suppose H contains a C_4-configuration X_1, X_2, Y_1, Y_2. Assume w.l.o.g. that $\bigcup_{U\in\mathcal{F}_1\cup\mathcal{F}_2} U \subseteq P_i$ for some $i \leq r$. ($\mathcal{F}_1$ and $\mathcal{F}_2$ are as defined in Definition 5.2.) Suppose that we can add Φ edges to H such that the new graph H' becomes k-edge-connected and no new edge is added inside some $P_j \in \mathcal{P}$. Since $(k-d(X_1))+(k-d(X_2)) = \Phi$ our assumption shows that exactly one endvertex of each new edge is in $X_1 \cup X_2$. Hence no edge is added between X_1 and X_2. Furthermore, Definition 5.2 (iv) shows the sets X_1, Y_1, Y_2 all have degree k in H'. However, this contradicts Lemma 2.3. □

LEMMA 5.5. *If H contains a C_6-configuration, then it cannot be made k-edge-connected by adding $\Phi = 3$ edges without violating the partition constraints.*

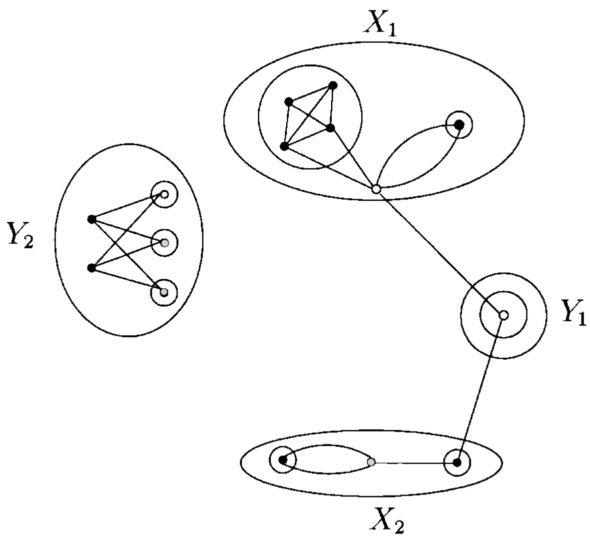


FIG. 3. *A graph $H = (V, E)$ with V partitioned into black, white, and shaded vertices. The sets X_1, X_2, Y_1, Y_2 form a C_4-configuration in H with respect to $k = 3$. Subpartitions $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}'_1, \mathcal{F}'_2$ are shown by inner circles.*

Proof. Suppose that $X_1, X_2, Y_1, Y_2, Z_1, Z_2$ form a C_6-configuration in H and we can make H k-edge-connected by adding three edges. Then each of the sets $X_1, X_2, Y_1, Y_2, Z_1, Z_2$ will be incident with one new edge. Since we use only three new edges, none of which is added inside some class of $\mathcal{P}$, it is easy to see that one of the new edges goes between two consecutive sets, say w.l.o.g. between X_1 and Y_1. Then $d(X_1 \cup Y_1) = k-1$ in the new graph, a contradiction. ∎

For the next two lemmas recall that $G = (V + s, E \cup F)$ is the graph obtained after executing Steps 1 and 2 of the algorithmic proof of Lemma 4.1.

LEMMA 5.6. *Suppose $A_1, B_1, C_1, A_2, B_2, C_2$ is a C_6-obstacle in G that is not a C_6-configuration in H. Then we can replace one edge incident to s by a new edge incident to s so that the resulting graph satisfies* (1) *and has a complete allowed splitting.*

Proof. Let $S = \{a_1, a_2,\ b_1, b_2,\ c_1, c_2\}$ and $a_i \in A_i$, $b_i \in B_i$, $c_i \in C_i$ for $i = 1, 2$. Let P_a, P_b, P_c be the distinct classes of $\mathcal{P}$ such that $a_1, a_2 \in P_a$, $b_1, b_2 \in P_b$, and $c_1, c_2 \in P_c$. Let $A'_i \subseteq A_i, B'_i \subseteq B_i, C'_i \subseteq C_i$, $i = 1, 2$ be minimal subsets of degree $k-1$ in $G - s$. Since $d(s) = 6$, our remark at the end of section 4 shows $\Phi = d(s)/2 = 3$. Now since the given partition is not a C_6-configuration, we can assume w.l.o.g. that $A'_1 \cap P_d \neq \emptyset$ for some $d \neq a$. Let $x \in P_d \cap A'_1$ and replace the edge sa_1 by the edge sx. Denote the new graph by G'. This graph satisfies (1) by Lemma 2.11. G' does not contain a C_6-obstacle since s has exactly one neighbor in P_a.

Suppose G' contains a C_4-obstacle $A^*_1, A^*_2, B^*_1, B^*_2$, where the labeling is chosen so that s has only neighbors from one $P_i \in \mathcal{P}$ in $A^*_1 \cup A^*_2$. Then by the definition of a C_4-obstacle, s has three neighbors in P_i. Hence x belongs to both $A^*_1 \cup A^*_2$ and $P_b \cup P_c$, say w.l.o.g. $x \in A^*_1 \cap P_b$. By Lemma 2.5 none of the sets $A^*_1, A^*_2, B^*_1, B^*_2$ forms a crossing pair with any of the sets $A_1, B_1, C_1, A_2, B_2, C_2$. There are no two sets $Z \in \{A_1, B_1, C_1, A_2, B_2, C_2\}$ and $W \in \{A^*_1, A^*_2, B^*_1, B^*_2\}$ such that $Z \cup W = V$

since Z contains only one neighbor of s. This implies that $A_1^* \cup A_2^* = A_1 \cup B_1 \cup B_2$. But Lemma 2.4 implies this is impossible since all five sets have degree k. Hence G' contains no C_4-obstacle.

Now Theorem 3.6 shows G' has a complete allowed splitting at s. ∎

We say that a subpartition $\mathcal{F}$ of V is *irreducible* if every $X \in \mathcal{F}$ has $d(X) < k$.

LEMMA 5.7. *Suppose A_1, B_1, A_2, B_2 is a C_4-obstacle in G that is not a C_4-configuration in H. Then either we can replace one edge incident to s by a new edge, or we can replace two edges incident to s by two new edges, so that the resulting graph satisfies* (1), *has no C_4-obstacle, and satisfies $d(s, P_i) \le d(s)/2$ for all $1 \le i \le r$.*

Proof. Choose the labeling of A_1, B_1, A_2, B_2 so all neighbors of s in $A_1 \cup A_2$ belong to one P_i. Let $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_1', \mathcal{F}_2'$ be arbitrary irreducible subpartitions of A_1, A_2, B_1, B_2, respectively, such that in H, $k - d(A_1) = \sum_{U \in \mathcal{F}_1}(k - d(U))$, $k - d(A_2) = \sum_{U \in \mathcal{F}_2}(k - d(U))$, $k - d(B_1) = \sum_{W \in \mathcal{F}_1'}(k - d(W))$, and $k - d(B_2) = \sum_{W \in \mathcal{F}_2'}(k - d(W))$. Our remark at the end of section 4 shows that in G, $d(s)/2 = \Phi$. Now since A_1, A_2, B_1, B_2 is not a C_4-configuration in H we must have that $\bigcup_{U \in \mathcal{F}_1 \cup \mathcal{F}_2} U$ contains a vertex $x \notin P_i$, say w.l.o.g. $x \in U \subseteq A_1$, $U \in \mathcal{F}_1$. Since $\mathcal{F}_1$ is irreducible, s has a neighbor $u \in U \cap P_i$. Replace the edge su by the new edge sx. If all neighbors of s in $B_1 \cup B_2$ belong to some P_j, then since A_1, A_2, B_1, B_2 do not form a C_4-configuration in H it follows again that $\bigcup_{W \in \mathcal{F}_1' \cup \mathcal{F}_2'} W$ contains a vertex $y \notin P_j$, say w.l.o.g. $y \in W \subseteq B_1$, $W \in \mathcal{F}_1'$. The irreducibility of $\mathcal{F}_1'$ shows s has a neighbor $v \in W \cap P_j$. Replace the edge sv by the new edge sy.

Denote the graph resulting from these edge replacements by G'. G' satisfies (1) by Lemma 2.11 and satisfies $d(s, P_i) \le d(s)/2$ for all $1 \le i \le r$. We will show G' contains no C_4-obstacle. (It may contain a C_6-obstacle.) Suppose G' contains a C_4-obstacle $A_1^*, A_2^*, B_1^*, B_2^*$ where the labeling is chosen so all neighbors of s in $A_1^* \cup A_2^*$ belong to one class of $\mathcal{P}$. An argument similar to those in section 3 shows that the two obstacle partitions coincide. (Specifically, Lemma 2.5 implies that no two obstacle sets cross. No two obstacle sets cover V since one would violate Lemma 2.8. Lemma 2.4 completes the argument.)

Observe that $A_1^* \cup A_2^*$ is not the set $A_1 \cup A_2$ or $B_1 \cup B_2$, by our edge replacements. Thus

$$A_1^* \cup A_2^* = A_p \cup B_q$$

for some indices p, q. We claim that $d(s, A_p),\ d(s, B_q) > 1$. Suppose $d(s, A_p) = 1$ (the argument for $d(s, B_q) = 1$ is similar). Write $\overline{p} = 3 - p, \overline{q} = 3 - q$. Then $d(s, A_{\overline{p}}) = d(s)/2 - 1$, and so $d(s, B_{\overline{q}}) = 1$. The definition of the two C_4-obstacles shows no edge of H leaves $A_p \cup B_{\overline{q}}$. (An edge leaving A_p goes to $A_{\overline{p}} \cup B_{\overline{q}}$ by the second obstacle and to $B_q \cup B_{\overline{q}}$ by the first obstacle. This is similar for an edge leaving $B_{\overline{q}}$.) Thus $d(A_p \cup B_{\overline{q}}) = 2$, contradicting $k \ge 3$.

All neighbors of s in $A_1^* \cup A_2^*$ are in the same partition class. Thus if $p = 1$, then $d(s, A_1) = 1$ (by our edge replacement). If $p = 2$, then $q = 1$ and $d(s, B_q) = 1$ (since no edge was replaced in A_2). Both cases give a contradiction. We conclude the C_4-obstacle does not exist. ∎

The following result holds for k of arbitrary parity. It subsumes Lemma 4.1.

THEOREM 5.8. *Let $k \ge 2$ and let $H = (V, E)$ be a graph with a partition $\mathcal{P} = \{P_1, \ldots, P_r\}$, $r \ge 2$ of V. Then $OPT_{\mathcal{P}}^k(H) = \Phi$ unless H contains a C_4- or C_6-configuration, in which case $OPT_{\mathcal{P}}^k(H) = \Phi + 1$.*

Proof. We give a constructive proof. (The same outline is followed by our efficient algorithm.) Start by performing Steps 1 and 2 of the procedure of section 4, and let

G be the resulting graph. As remarked at the end of section 4, G has $d(s)/2 = \Phi$, and also $d(s, P_i) \leq d(s)/2$ for all $1 \leq i \leq r$. Now execute the procedure given after Lemma 3.4 to split off edges of G. As remarked, eventually we obtain either a complete allowed splitting or a graph with $d(s) = 4$ containing a C_4-obstacle A_1, A_2, B_1, B_2. In the first alternative $OPT^k_{\mathcal{P}}(H) = \Phi$. The rest of the proof concerns the second alternative. We distinguish two cases depending on how the new edges were added to G during the sequence of splittings.

Case I. At least one split added an edge inside one of the sets A_1, A_2, B_1, B_2.

Say w.l.o.g. that an edge was added inside A_1 when we split off the pair sx, sy. Undoing this split gives a graph G' that has $d(s) = 6$ and still satisfies (1). If neither x nor y belongs to a partition P_i with $d(s, P_i)$ maximum in G', then it is easy to see that G' does not contain an obstacle. Thus Theorem 3.6 shows G' has a complete allowed splitting and $OPT^k_{\mathcal{P}}(H) = \Phi$.

In the opposite case G', x and y have all the properties of G, x, and y in Case 1 and Case A of the proof of Theorem 3.6. Following Case 1 we define another allowed splitting sx', sy' for which there are two possibilities. The first possibility is that splitting off sx', sy' gives a graph with no obstacle. Since this graph has $d(s) = 4$, Lemma 3.4 shows the allowed splitting can be completed. Thus $OPT^k_{\mathcal{P}}(H) = \Phi$.

Lemma 3.9 shows that the second possibility is that graph G' contains a C_6-obstacle. Suppose first that $G' \neq G$, i.e., $d(s) \geq 8$ in G. Let G' be the result of executing a split in the graph G''. Lemma 3.5 shows G'' has a splitting that results in a graph G^*, where $d(s) = 6$ and there is no C_4-obstacle and no C_6-obstacle. Theorem 3.6 shows we can find a complete allowed splitting in G^*. Thus $OPT^k_{\mathcal{P}}(H) = \Phi$.

The remaining case is when $G' = G$, so G has a C_6-obstacle. Lemma 5.6 shows there are two possibilities. The first possibility is that we can repair this C_6-obstacle by moving one edge incident with s, to get a graph with a complete allowed splitting. This shows $OPT^k_{\mathcal{P}}(H) = \Phi$. The second possibility is that H has a C_6-configuration. In this case it follows from Lemmas 5.5 and 5.1 that $OPT^k_{\mathcal{P}}(H) = \Phi + 1$.

Case II. All splits added edges between different sets of the partition A_1, A_2, B_1, B_2.

In this case A_1, A_2, B_1, B_2 is a C_4-obstacle in G. Lemma 5.7 shows there are two possiblities. The first possibility is that A_1, A_2, B_1, B_2 is a C_4-configuration, in which case Lemmas 5.4 and 5.1 show $OPT^k_{\mathcal{P}}(H) = \Phi + 1$. The second possiblity is that we can move at most two edges incident with s so that the resulting graph G' has no C_4-obstacle. If G' has no C_6-obstacle either, then Theorem 3.6 shows $OPT^k_{\mathcal{P}}(H) = \Phi$. If G' has a C_6-obstacle, then the theorem holds as shown above. ∎

COROLLARY 5.9. *Let $k \geq 2$ and let $H = (V, E)$ be a graph with a partition $\mathcal{P} = \{P_1, \ldots, P_r\}$, $r \geq 2$ of V. If $OPT^k(H) \geq 2k + 1$, then $OPT^k_{\mathcal{P}}(H) = \Phi$.*

Proof. If H has a C_6-configuration, then $OPT^k(H) = 3$, so the corollary is vacuous. If H contains a C_4-configuration, then Definition 5.2(iv) shows $OPT^k(H) = \Phi \leq 2k$. ∎

COROLLARY 5.10. *Let $k \geq 2$, let $H = (V, E)$ be a bipartite graph with bipartition $V = A \cup B$, and let $\mathcal{P} = \{A, B\}$.*

- *Let* $\alpha = \max\left\{\lceil \frac{1}{2} \sum_{X \in \mathcal{F}} (k - d(X)) \rceil : \mathcal{F} \text{ a subpartition of } V\right\}$.
- *Let* $\beta'_1 = \sum_{v \in A} \max\{0, k - d(v)\}$.
- *Let* $\beta'_2 = \sum_{v \in B} \max\{0, k - d(v)\}$.
- *Let* $\Phi' = \max\{\alpha, \beta'_1, \beta'_2\}$.

Then $OPT^k_{\mathcal{P}}(H) = \Phi'$ unless k is odd and H contains a C_4-configuration, in which case $OPT^k_{\mathcal{P}}(H) = \Phi' + 1$. ∎

If $r = |V|$, we obtain the min-max equality of the unconstrained problem by Cai and Sun [3].

COROLLARY 5.11 (see [3]). *Let $H = (V, E)$ be an undirected graph and $k \geq 2$.*

- *Let $\alpha = \max\left\{\lceil \frac{1}{2}\sum_{X\in\mathcal{F}}(k - d(X))\rceil : \mathcal{F}$ a subpartition of $V\right\}$.*

Then $OPT^k(H) = \alpha$. □

6. Efficient algorithms. This section presents a strongly polynomial algorithm for edge connectivity augmentation with partition constraints. The time bound is $O(n(m + n\log n)\log n)$, the same as the best-known strong polynomial bound for the problem without partition constraints. Throughout this section n and m denote the number of vertices and edges in the given graph, respectively. Each edge xy has a given integer capacity, which in our notation equals $d(x, y)$. We represent splits by providing the edges sx, sy and an integral *multiplicity* γ, which indicates that the split is to be performed γ times (producing γ new copies of the edge xy).

Although our main concern is graphs that have arbitrary capacity functions, we also state time bounds for the special case of unit capacity graphs. In these graphs each edge has capacity one. Parallel edges are allowed, but each copy of an edge contributes one to the number of edges m. For unit capacity input graphs our algorithm runs in time $O(nm\log n)$, the best-known bound for the unconstrained problem on these graphs. (The output graph is allowed to contain high-capacity edges.) Throughout this section time bounds that apply to unit capacity graphs are explicitly designated as such; time bounds without such an explicit designation apply to arbitrary capacitated graphs.

Say the split sx, sy *joins* vertices x and y. Throughout this section it is convenient to define

$$h = \frac{k-1}{2}.$$

6.1. Basic facts. Assume $G = (V + s, E)$ has $d(s)$ even and satisfies (1). Consider a set $X \subseteq V$. G/X denotes the graph G with X contracted to a single vertex. A split sx, sy in G corresponds to a split in G/X in the obvious way. Set X is *contractible* if any collection of splits in G that is admissible in G/X is admissible in G. (Recall that a collection of splits is admissible if executing the splits results in a graph satisfying (1).)

We will generalize the following well-known fact first proved by Mader [15].

PROPOSITION 6.1. *A critical set is contractible.*

Proof. Consider a critical set C and a collection of splits in G that is admissible in G/C. Let G' be the result of executing the splits in G. We must show that any nonempty set $X \subset V$ satisfies $d'(X) \geq k$ (as usual d' denotes the degree function in G').

Case 1. $X \subseteq C$. No split joins two vertices of C, since $d(C) = k$. Thus the degree of any subset of C is preserved by the splitting, so $d'(X) \geq k$.

Case 2. $C \subseteq X$ or $X \cap C = \emptyset$. Since the splitting is admissible in G/C, $d'(X) \geq k$.

Case 3. Neither Case 1 nor Case 2 applies.

By (3),

$$d'(C) + d'(X) \geq d'(C - X) + d'(X - C).$$

Note that both sets $C - X$ and $X - C$ are nonempty. The analysis of Case 1 shows $d'(C - X) \geq k = d'(C)$. Case 2 shows $d'(X - C) \geq k$. Substituting these relations in the displayed inequality gives the desired conclusion $d'(X) \geq k$. □

To generalize Proposition 6.1 define a *k-bisection* of a set $C \subseteq V$ to be a partition of C into exactly two sets of degree k.

LEMMA 6.2. *A dangerous set having no k-bisection is contractible.*

Proof. The argument is similar to Proposition 6.1. Consider a set C with $d(C) = k+1$ and a collection of splits in G that is admissible in G/C. Taking G' and d' as in Proposition 6.1, we must show that any nonempty set $X \subset V$ satisfies $d'(X) \geq k$.

If X satisfies Case 1 or Case 2 of Proposition 6.1, we have $d'(X) \geq k$. (For Case 1 we use the fact that no split joins vertices of C since $d(C) = k+1$.) If neither Case 1 nor Case 2 applies, then $C - X$ and $X \cap C$ are both nonempty. Since C has no k-bisection, at least one of these sets has degree $\geq k+1 = d'(C)$. If $d'(C-X) \geq d'(C)$ we argue as in Case 3 of Proposition 6.1. This leaves only the following case.

Case 4. $d'(C \cap X) \geq d'(C)$. First suppose $C \cup X \neq V$. By (2)

$$d'(C) + d'(X) \geq d'(C \cap X) + d'(X \cup C).$$

Case 2 shows $d'(X \cup C) \geq k$. Substituting this and the inequality for Case 4 gives $d'(X) \geq k$.

Next suppose $C \cup X = V$. Applying Lemma 2.8 in G/C to C shows $d'(s, X) \geq d'(s, X - C) \geq d'(s, C) \geq d'(s, C - X)$. Thus $d'(X) \geq d'(C - X)$. Case 1 shows $d'(C - X) \geq k$, so $d'(X) \geq k$ as desired. ◻

LEMMA 6.3. *Suppose $d(C) = k+1$ and C has a k-bisection. Let the sets of the k-bisection be A and B. Then*

(a) *k is an odd integer;*

(b) *A and B are both maximal degree k subsets of C, and C has no other maximal degree k subsets;*

(c) *Sets A and B form the unique minimum cut of graph $G[C]$. This cut has value $d(A, B) = h$.*

Proof.

(a) Applying (2) shows $2k = d(A) + d(B) = d(C) + 2d(A, B)$. This implies $d(C) = k+1$ is even.

(b) Let M be a maximal subset of C having degree k. Assume M is distinct from A and B. We will deduce a contradiction, thereby proving (b).

M is not a subset of A or B. Since $d(M) = k$, $M \subset C$. Thus M is crossing with at least one of the sets A, B, say A. Since A and M are crossing critical sets, their union is critical by (2). This contradicts the maximality of M.

(c) Consider any cut of $G[C]$, say a partition of C into sets X and Y. Applying (2), $2k \leq d(X) + d(Y) = d(X \cup Y) + 2d(X, Y)$. Since $d(X \cup Y) = k+1$, this implies $d(X, Y) \geq \frac{k-1}{2} = h$. Equality holds throughout for the cut A, B. Furthermore, (b) shows that A, B is the unique k-bisection of C, so any other cut X, Y has $2k < d(X) + d(Y)$, whence $d(X, Y) > h$. This proves (c). ◻

We conclude this section by mentioning two simple properties of splittings that will be useful. As before let G satisfy (1) with $d(s)$ even. The first property is that any vertex x in a dangerous set has $d(s, x) \leq d(s)/2$ (equivalently, $d(s, x) \leq d(s, V - x)$). This follows from Lemma 2.8.

For the second property suppose that every vertex x has $d(s, x) \leq d(s)/2$ and, furthermore, there is a unique complete splitting at s such that no split joins a vertex to itself. Then this splitting is admissible. To see this recall that a complete admissible splitting exists (Theorem 2.12(a)). If it contains a split sx, sx it also contains a split sy, sz with $y, z \neq x$ (since $d(s, x) \leq d(s)/2$). Replacing these two splits by sx, sy and sx, sz results in another admissible splitting. Performing this replacement as many times as possible results in the unique complete splitting with no split joining a vertex to itself, so this splitting is admissible.

6.2. The splitting algorithm for even k. Consider the problem of finding a complete admissible splitting at s. We present an algorithm that runs in time $O(n(m+n\log n)\log n_s)$. Here n_s is the number of neighbors of s, $n_s \le n$. Our algorithm is a modification of the algorithm of Nagamochi, Nakamura, and Ibaraki [18], using contraction as the main operation. The main advantage of our approach is that it can be extended to find a complete allowed splitting for odd k. (Both algorithms can be so extended for even k.) In addition there is a slight efficiency advantage: the algorithm of [18] finds a complete splitting with $O(n_s \log n_s)$ distinct splits. Our splitting contains $O(n_s)$ distinct splits, the asymptotically optimum number. This implies that fewer distinct edges get added to the graph. It improves the space bound, from $O(m + n_s \log n_s)$ to $O(m)$. Also, while the time bound of [18] for k even is the same as our bound, for k odd it is our bound with $\log n_s$ changed to $\log n$ (so in some cases our algorithm is slightly faster). This section presents our splitting algorithm for even k.

We first summarize the algorithm of Nagamochi, Nakamura, and Ibaraki [18] (see also [17]). It is based on two routines. The first routine, C-SPLIT, finds a complete splitting at s. This splitting need not be admissible. However, it has the following property. C-SPLIT is called with a subpartition of V whose sets contain all neighbors of s. No split found by C-SPLIT joins two vertices in the same partition set.

The second routine, HOOK-UP, is called with a complete splitting. It identifies a maximal subset of these given splits that is admissible. It executes these splits on the graph (so (1) is preserved). It also returns a family $\mathcal{Y}$ of disjoint dangerous sets, such that each given split that was not executed joins two vertices in the same dangerous set of $\mathcal{Y}$. (In terms of our representation of splits using multiplicities, the maximal admissible subset of splits is an assignment of a new multiplicity γ' to each given split sx, sy of multiplicity γ, with $\gamma' \le \gamma$. Increasing any value γ' destroys admissiblity. If $\gamma' < \gamma$, then x and y are in the same set of $\mathcal{Y}$.) HOOK-UP runs in time $O(n(m + n\log n))$. (The time is $O(nm)$ on a unit capacity graph.)

The algorithm of [18] for even k is as follows. The algorithm is a loop that terminates when $d(s) = 0$, i.e., vertex s has been completely split off. Each iteration of the loop starts by using C-SPLIT to find a complete splitting, with no split joining two vertices in the same set of $\mathcal{Y}$. (For the first iteration $\mathcal{Y}$ is initialized to the family of singleton sets $\{x\}$ where x ranges over all neighbors of s. After the first iteration, $\mathcal{Y}$ was set by HOOK-UP in the previous iteration.) HOOK-UP is called to execute a maximal admissible subset of these splits and to update $\mathcal{Y}$ for these splits. If $d(s) = 0$, then a complete admissible splitting at s has been executed, so the algorithm halts.

It is proved in [18] that in each iteration the size of the family $\mathcal{Y}$ is halved. This implies there are $O(\log n_s)$ iterations. Each iteration uses time $O(n(m + n\log n))$, so the total time is $O(n(m + n\log n)\log n_s)$.

We now present our modified splitting algorithm for even k. Our approach is always to contract the sets of $\mathcal{Y}$. Instead of C-SPLIT we use the following algorithm E-SPLIT to find a complete splitting. This algorithm seems easier to modify for the case of odd k and the case of allowed splittings. E-SPLIT outputs a complete splitting at s, with no split joining a vertex to itself (this specification differs slightly from C-SPLIT). Let $x_i, i = 1, \ldots, p$ be the distinct neighbors of s, indexed so that $d(s, x_1)$

equals the maximum degree $d(s, x_i)$. E-SPLIT is always called with

$$d(s, x_1) \leq \sum_{i>1} d(s, x_i).$$

E-SPLIT maintains d_i as the value of $d(s, x_i)$ if the splits output so far were to be executed. So initially $d_i = d(s, x_i)$ for $i = 1, \ldots, p$. E-SPLIT maintains the preceding inequality throughout its execution, i.e.,

$$d_1 \leq \sum_{i>1} d_i. \tag{7}$$

Procedure E-SPLIT.

Step 1. While (7) holds with strict inequality, repeat the following.

Step 1(a). Choose distinct indices $i, i' > 1$ with both $d_i, d_{i'} > 0$. Output split $sx_i, sx_{i'}$ with multiplicity $\gamma = \min\{d_i, d_{i'}, (\sum_{i>1} d_i - d_1)/2\}$. Decrease d_i and $d_{i'}$ by γ.

Step 2. (At this point (7) holds with equality.) Output splits sx_1, sx_i with multiplicity d_i, for all indices $i > 1$ that have positive d_i.

We show that E-SPLIT works correctly, i.e., it outputs a complete splitting at s, with no split joining a vertex to itself. First observe that $d_1 = \max\{d_i : i \geq 1\}$ throughout the algorithm. This follows since Step 1(a) only decreases values d_i with $i > 1$. Also observe that (7) holds throughout the algorithm, by the definition of γ and the termination condition of Step 1(a).

Now consider Step 1. First observe there are at least two indices $i > 1$ with d_i positive, since (7) holds with strict inequality and every d_i, $i > 1$ is no larger than d_1.

Next observe that Step 1 eventually halts. This follows because the sum $\sum_{i>1} d_i$ has the same parity as d_1 when E-SPLIT starts (since $d(s)$ is even). Each split maintains this relation, since it decreases the sum by an even integer. Thus eventually (7) holds with equality and Step 1 completes.

Step 2 works correctly since (7) holds with equality.

To analyze the efficiency of E-SPLIT, first observe that it outputs at most p distinct splits. This follows because, with one exception, every split (in Step 1 or Step 2) decreases some d_i, $i > 1$ to zero. The exception (which may not exist) is the unique split in Step 1 that achieves equality in (7).

Now it is easy to see that E-SPLIT uses total time $O(p)$.

Our splitting algorithm works by finding splits in a contracted graph H. We execute these splits in both the contracted graph H and the original graph G. (So both G and H change throughout the algorithm. The vertex set of G is always $V + s$.) The following invariant holds throughout the algorithm:

Graph H can be obtained from G by contracting a collection of disjoint dangerous subsets of V. Both G and H satisfy (1).

The invariant will guarantee correctness of the algorithm.

We use the following auxiliary routine to translate a split from H to G. It is called with two disjoint sets $X, Y \subset V$ and an integer γ. X and Y have been contracted to vertices in H, and the split sX, sY with multiplicity γ is admissible, in H.

Procedure TRANSLATE(X, Y, γ).

Repeat Step 1 until $\gamma = 0$.

Step 1. Choose a vertex x of G corresponding to X: if X is a vertex of G let $x = X$; otherwise choose a vertex $x \in X$ that is a neighbor of s in G. Similarly choose a vertex y of G corresponding to Y. Execute split sx, sy in G with multiplicity

$\gamma_0 = \min\{d(s,x), d(s,y), \gamma\}$ (here d is the degree function in G). Decrease γ by γ_0.

Assuming the invariant holds, TRANSLATE executes a sequence of splits that is admissible in G. This follows because any dangerous set is contractible when k is even (by Proposition 6.1, Lemma 6.2, and Lemma 6.3(a)). The final G obtained by TRANSLATE corresponds to H with split sX, sY executed (so the invariant is preserved).

Now we present our splitting algorithm for even k. We assume the given graph G satisfies (1) with $d(s)$ even. In addition we assume that every vertex x satisfies $d(s,x) \leq d(s)/2$. This inequality holds automatically in many applications of splitting, e.g., Frank's algorithm of section 2. If not, we can enforce the inequality by deleting all but $d(s) - d(x)$ edges sx for the vertex x satisfying $d(s,x) > d(s)/2$. (To prove this note that in a complete admissible splitting at most $d(s) - d(x)$ copies of edge sx are split off with an edge sy, $y \neq x$, so the remaining copies are in splits sx, sx which do not affect the connectivity.)

The splitting algorithm works simultaneously on graphs G and H that satisfy the invariant stated above. The algorithm executes a complete admissible splitting at s on the original graph G. The algorithm starts with H initialized to the given graph G.

Procedure EVEN-SPLIT.

Repeat Steps 1–3 until $d(s) = 0$.

Step 1. Call E-SPLIT to find a complete splitting at s in graph H, with no split joining a vertex to itself.

Step 2. Call HOOK-UP to execute a maximal admissible subset of these splits in H and to return the corresponding family of dangerous sets $\mathcal{Y}$. For each split that was executed in H call TRANSLATE to execute it in G.

Step 3. In H contract each set $Y \in \mathcal{Y}$ to a single vertex.

To show EVEN-SPLIT is correct first observe that Step 1 works correctly because (7) holds whenever E-SPLIT is called. To prove (7) recall that it holds by assumption when the algorithm starts and E-SPLIT is called for the first time. In subsequent iterations Step 3 ensures that every vertex of H adjacent to s forms a dangerous set. This implies (7) (by the first remark at the end of section 6.1).

Next observe that the invariant is preserved by all steps: the splits executed by HOOK-UP and TRANSLATE preserve the invariant (by the discussion of TRANSLATE). The contractions done in Step 3 clearly preserve the invariant.

The invariant implies that if EVEN-SPLIT halts, it has executed a complete admissible splitting on the original graph G. Thus we need only prove termination. We do this by showing there are $O(\log n_s)$ iterations in EVEN-SPLIT.

Consider graph H after Step 2. Any remaining neighbor x of s was in an unexecuted split sx, sy, with x and y now belonging to the same set of $\mathcal{Y}$. This set gets contracted in Step 3. Now let s_i denote the number of distinct neighbors of s at the start of the ith iteration. Then

$$s_{i+1} \leq \frac{s_i}{2}.$$

Since $s_1 = n_s$, there are $O(\log n_s)$ iterations.

Next we show that the total number of distinct splits executed in Step 2 in all iterations is $O(n_s)$, for both G and H. This fact for graph H is important for reasons of efficiency: each distinct split executed in Step 2 can add a new edge to H, increasing space and the time for subsequent computations. The fact for graph G is a measure of quality of the complete splitting produced by EVEN-SPLIT.

To prove the fact for H, Step 1 (of EVEN-SPLIT) gives a splitting with at most s_i distinct splits, by our discussion of E-SPLIT. So clearly Step 2 executes at most s_i distinct splits in H. The inequality displayed above shows the sum of all quantities s_i is $O(s_1) = O(n_s)$.

To prove the fact for G it suffices to show that a total of at most n_s more splits is done in G than in H. This follows because for a given split sX, sY in H, each execution of Step 1 of TRANSLATE excepting the last decreases a value $d(s,x)$ to 0.

Now we can show the time for the entire algorithm EVEN-SPLIT is $O(n(m + n\log n)\log n_s)$. Graphs G and H both have $O(m + n_s) = O(m)$ edges. Each call to HOOK-UP takes time $O(n(m + n\log n))$. The remaining processing in EVEN-SPLIT takes time $O(m)$ in each iteration. There are $O(\log n_s)$ iterations, so the desired time bound follows. It is easy to show that for unit capacity graphs EVEN-SPLIT runs in time $O(nm \log n_s)$. The argument is the same as above, with the observation that G and H always have $O(m)$ edges because each split decreases the number of edges.

6.3. The splitting algorithm for odd *k*. We turn to our splitting algorithm for k an odd integer. The algorithm of [18] is based on the (strong) successive augmentation property [4]: it first executes splits to achieve connectivity $k - 1$, and then it augments the connectivity of the resulting graph to k. Our algorithm is a modification of the algorithm for k even, again based on contraction. We present the algorithm by specifying the changes made to the k even algorithm. The resulting algorithm will apply to both even and odd k.

The main change in the algorithm is to avoid creating splits that are obviously wrong. Call a split sx, sy *foolish* if $\{x\}$ and $\{y\}$ are critical sets and $d(x,y) = h$. A foolish split is inadmissible, since $d(\{x,y\}) = d(x)+d(y)-2d(x,y) = 2k-2h = k+1$. We modify E-SPLIT so it creates at most C foolish splits for some constant C. (We do not take care to minimize the value of C. Also the value of C will change when we modify the algorithm to do allowed splitting.)

We start with two observations about foolish splits. First, a vertex x is in at most two foolish splits. This follows because $\{x\}$ must be critical and x must be a neighbor of s, i.e., $d(x) = k = 2h + 1$ and $d(s,x) \geq 1$. Thus $d(x,y) = h$ for at most two vertices y.

Second, in a complete splitting a vertex x is in at most one foolish split. This is obvious if $d(s,x) = 1$. If $d(s,x) > 1$, then the above calculation shows we can have $d(x,y) = h$ for at most one vertex y.

Now we present procedure G-SPLIT. Like E-SPLIT, it outputs a complete splitting with no split joining a vertex to itself. G-SPLIT is the same as E-SPLIT with one change. Step 1(a) chooses indices i, i' by executing the following Choice Step.

Choice Step. Choose an index $i > 1$ with $d_i > 0$. Let F be the set of indices $f > 1$ with sx_i, sx_f a foolish split. (Note $|F| \leq 2$.) Choose an index $i' \neq i, 1$ with $d_{i'} > 0$ and, if possible, $i' \notin F$. If $i' \in F$ and no splits have been output yet in Step 1, redefine i and i' to be the two indices of F.

Note that if k is even there are no foolish splits, so G-SPLIT is the same as E-SPLIT. It is not immediately obvious that the last statement of the Choice Step is well defined or correct. We now prove the algorithm works as desired. Assume that G-SPLIT (like E-SPLIT) is called with values d_i corresponding to vertices in a graph H that satisfies (1).

LEMMA 6.4.

(a) *G-SPLIT outputs a complete splitting, with no split joining a vertex to itself.*

(b) *G-SPLIT outputs at most $C = 2$ foolish splits.*

(c) *The first split output by G-SPLIT is not foolish.*

Proof. Regarding part (a), by definition G-SPLIT never outputs a split joining a vertex to itself. Now consider Step 2 of G-SPLIT. It outputs at most one foolish split, since all its splits involve vertex x_1. Suppose no splits are output in Step 1. This means G-SPLIT starts with equality in (7). Step 2 outputs the unique complete admissible splitting of H with no split joining a vertex to itself (by the second remark at the end of section 6.1). Clearly (c) holds in this case.

To complete the proof we analyze Step 1 of G-SPLIT. For (b) it suffices to show Step 1 outputs at most one foolish split. We consider two cases.

Case 1. For the first value of i chosen in Step 1, $i' \notin F$.

This certainly implies part (c). Furthermore, it ensures that the last statement of the Choice Step never redefines i and i'. Now suppose at some point Step 1 outputs a foolish split. Right before this occurs, at most three vertices x_j have $j > 1$ and $d_j > 0$ (specifically the vertices x_i and $x_{i'}$ in the foolish split, and at most one other vertex x_f with $f \in F$). Step 1 can output at most one foolish split involving only these three vertices (since a given vertex is in at most one foolish split that gets output). Thus (b) holds, as well as (a).

Case 2. Case 1 does not apply.

Let I be the set of all indices that are larger than 1 with d_i initially positive. A complete admissible splitting exists. Recall (7) holds with strict inequality when Step 1 is executed. This implies there is an admissible split $sx_i, sx_{i'}$ with $i, i' \in I$.

Suppose $|I| = 2$. Obviously the first value of i gives the only split that is possible. This split is not foolish since it is admissible. Thus Case 1 applies.

Now suppose $|I| \geq 3$. The first value of i gives a set with $F = I - \{i\}$. Thus $|I| \leq 3$ and so $|I| = 3$. Thus the unique admissible split formed from indices in I comes from the two indices in F. The last statement of the Choice Step outputs this split. This finishes Step 1 (again since a complete admissible splitting exists). Thus (a)–(c) hold. ◻

G-SPLIT outputs at most p distinct splits, since it is a special case of E-SPLIT. It is easy to see that G-SPLIT runs in time $O(m + n)$.

Now we present the splitting algorithm GENERAL-SPLIT. This procedure constructs a complete admissible splitting for arbitrary k. As before we assume the given graph G satisfies (1) with $d(s)$ even, and every vertex x satisfies $d(s, x) \leq d(s)/2$. GENERAL-SPLIT is derived from EVEN-SPLIT by making two changes. Step 1 of GENERAL-SPLIT calls G-SPLIT rather than E-SPLIT. Step 3, which does the contractions, is replaced by the following step.

Contraction Step. In H do the following for each set $Y \in \mathcal{Y}$: If Y is critical, contract Y. Otherwise ($d(Y) = k + 1$) test if Y has a k-bisection. If it does not, contract Y. If it does, say into sets A and B, contract A and contract B.

We make several remarks on the Contraction Step. If k is even, there are no k-bisections. Hence GENERAL-SPLIT becomes identical to EVEN-SPLIT. If Y has a k-bisection, after contracting A and B a split sA, sB is foolish. And if sx, sy is a foolish split giving a set $Y = \{x, y\}$, the Contraction Step does not change the graph when it processes Y.

The Contraction Step tests if Y has a k-bisection by using the algorithm of [16] to find a minimum cut of $H[Y]$. Lemma 6.3(c) shows this cut is the k-bisection, if one exists. It is a simple matter to check if the cut is a k-bisection.

We proceed to analyze GENERAL-SPLIT. To prove correctness we first state the new invariant:

Graph H can be obtained from G by contracting a collection of disjoint subsets of V, each of which is either critical or a dangerous set having no k-bisection. Both G and H satisfy (1).

The TRANSLATE routine preserves the new invariant. This follows by the argument of section 6.2 with one change: any set contracted in H is contractible in G. This follows from Proposition 6.1 and Lemma 6.2.

Now we can use the same argument as for EVEN-SPLIT to show that in Step 1 of GENERAL-SPLIT (7) holds and also GENERAL-SPLIT preserves its invariant. For the first argument note that the Contraction Step still ensures that after the first iteration every neighbor of s is in a dangerous set. For the second argument note that the Contraction Step clearly preserves the invariant.

We must also prove that GENERAL-SPLIT terminates. We do this by showing that in H, every iteration of GENERAL-SPLIT either decreases the number of neighbors of s or decreases the number of vertices. To do this recall the first split output by G-SPLIT is not foolish (Lemma 6.4(c)). Let this split be sx, sy with multiplicity γ.

First suppose HOOK-UP splits off γ pairs of edges sx, sy. We show this decreases the number of neighbors of s in H. This is certainly true if $\gamma = \min\{d(s,x), d(s,y)\}$. It is easy to see that when the remaining possibility for γ holds, sx, sy is the last (as well as the first) split output by Step 1 of G-SPLIT. Thus after executing this split equality holds in (7), and the remaining splits are the only ones possible in a complete admissible splitting at s. Hence these splits are all admissible. Thus this iteration of GENERAL-SPLIT executes a complete splitting, resulting in no remaining neighbors of s.

On the other hand suppose HOOK-UP splits off fewer than γ pairs of edges sx, sy. HOOK-UP returns a set $Y \in \mathcal{Y}$ containing both x and y, and $Y \neq \{x, y\}$ since the split is not foolish. Hence the Contraction Step decreases the number of vertices of H when it processes set Y. In summary we have shown that GENERAL-SPLIT always terminates.

Now we prove the time bound. The key step is to prove there are $O(\log n_s)$ iterations in GENERAL-SPLIT.

We first introduce some notation. Let H_i be the graph H at the start of the ith iteration of GENERAL-SPLIT. (Our argument will not refer to the corresponding graph G.) For example, H_1 is the given graph. The degree function in H_i is denoted d_i. Suppose v is a vertex in H_i. For $j \geq i$, $\overline{v}$ denotes the vertex in graph H_j that contains v. When we use this notation the value of j will be explicitly stated or clear from the context. For instance $d_j(\overline{v})$ denotes the degree in H_j of vertex $\overline{v}$ in H_j.

We classify the contractions made in Contraction Steps as follows. Call a contraction *good* if it reduces the number of vertices that contain a neighbor of s in H_1. In other words let X be a set that gets contracted in graph H_i. In a good contraction X contains distinct vertices $\overline{v}$ and $\overline{w}$ (in H_i) where both v and w are neighbors of s in H_1. ($\overline{v}$ and $\overline{w}$ need not be neighbors of s in H_i.) A contraction that is not good is *bad*. It is easy to see that a contraction is good if $X \in \mathcal{Y}$. So in a bad contraction X is one of the two sets of a k-bisection.

In the following discussion assume the number of iterations is at least four. Thus graph H_i exists for $1 \leq i \leq 4$.

LEMMA 6.5. *Let v be a neighbor of s in H_1. At least one of the following alternatives holds for some index i, $1 \leq i \leq 3$:*

(i) $\overline{v}$ *is not a neighbor of* s *in* H_4;
(ii) $\overline{v}$ *is in a foolish split created in* H_i;
(iii) $\overline{v}$ *is in a good contraction done in* H_i;
(iv) w, *a neighbor of* s *in* H_i *with* $d_i(\overline{v}, w) = h$, *is in a good contraction done in* H_j, *for some* j *in* $i < j \leq 3$.

Proof. We assume none of alternatives (i)–(iv) hold and derive a contradiction.

Since (i) fails, $\overline{v}$ is a neighbor of s in H_4. Thus for $1 \leq i \leq 3$, $\overline{v}$ is in a split created in H_i but not executed by HOOK-UP. For $1 \leq i \leq 3$ let $s\overline{v}, sw_i$ be such a split. (Thus w_i is a vertex of H_i.) We will show that in H_4 vertices $\overline{v}, \overline{w}_1, \overline{w}_2$, and $\overline{w}_3$ are distinct and in addition satisfy inequality (8). This will lead to the desired contradiction.

Since (iii) fails, split $s\overline{v}, sw_i$ results in a bad contraction. This implies

$$d_{i+1}(\overline{v}) = k.$$

Also Lemma 6.3(c) shows

$$d_{i+1}(\overline{v}, \overline{w}_i) = h.$$

Again since (iii) fails, $\overline{v} \neq \overline{w}_i$ in H_4 for $1 \leq i \leq 3$. This plus the preceding equation implies that for $i < j \leq 4$,

$$d_j(\overline{v}, \overline{w}_i) \geq h. \tag{8}$$

Next we show that in H_4, $\overline{w}_i \neq \overline{w}_j$ for any indices i, j with $1 \leq i < j \leq 3$. First we show $\overline{w}_i \neq w_j$ in H_j. As noted above, $d_j(\overline{v}) = k$. Since (iv) fails, $d_j(\overline{w}_i) = k$. By (8), $d_j(\overline{v}, \overline{w}_i) \geq h$. If equality holds, then $\overline{w}_i \neq w_j$ because (ii) fails. If strict inequality holds, then $\overline{w}_i \neq w_j$ because $d_{j+1}(\overline{v}, \overline{w}_i) > h = d_{j+1}(\overline{v}, \overline{w}_j)$. We conclude that $\overline{w}_i \neq w_j$ in H_j.

Thus to have $\overline{w}_i = \overline{w}_j$ in H_4 a contraction must merge the distinct vertices $\overline{w}_i$ and $\overline{w}_j$ in some graph H_r, $r > j$. But since (iv) fails, this never occurs. Hence $\overline{w}_i \neq \overline{w}_j$ in H_4.

Now the distinctness of vertices $\overline{v}, \overline{w}_1, \overline{w}_2$, and $\overline{w}_3$ in H_4 and inequality (8) in H_4 imply

$$d_4(\overline{v}) \geq d_4(\overline{v}, \overline{w}_1) + d_4(\overline{v}, \overline{w}_2) + d_4(\overline{v}, \overline{w}_3) + d_4(\overline{v}, s) \geq 3h + 1 > 2h + 1 = k.$$

(Note that $h \geq 1$ since $k \geq 3$.) But as noted above, $d_4(\overline{v}) = k$. This contradiction proves the lemma. ∎

Let s_i denote the number of neighbors of s in H_i. Let $C' = 3C$. We will use the lemma to show

$$s_4 - C' \leq \frac{5}{6}(s_1 - C'). \tag{9}$$

In proof, let S be the set of all neighbors of s in H_1. Define the following quantities corresponding to the four cases of Lemma 6.5:

- g is the number of vertices of S that are in a good contraction in at least one H_i, $1 \leq i \leq 3$.
- n is the number of vertices of S, not counted in g, that are not neighbors of s in H_4. (This conflicts with the usual usage of n but will not cause confusion.)
- f is the number of vertices of S, not counted in g or n, in a foolish split in some H_i, $1 \leq i \leq 3$.

- r is the number of vertices of S not counted in g, n or f.

By definition $s_1 = g + n + f + r$.

First observe that the vertices counted in g belong to at most $g/2$ distinct vertices in H_4. This follows because a good contraction merges at least two vertices of S.

The vertices counted in n are no longer neighbors of s in H_4. Thus we deduce

$$s_4 \leq \frac{g}{2} + f + r.$$

Next we show

$$r \leq 2g.$$

Lemma 6.5(iv) applies to any vertex v that is counted in r. Thus some vertex w counted in g has $d_4(\overline{v}, \overline{w}) \geq h$. Any contraction results in a vertex of degree at most $k+1$. Hence $d_4(\overline{w}) \leq k+1$. This implies $\overline{w}$ has $d_4(\overline{v}, \overline{w}) \geq h$ for at most four vertices v (since $5h \geq 2h+3 = k+2$). Thus $r \leq 4(g/2) = 2g$ as desired.

Finally recall that each iteration creates at most C foolish splits. Hence three iterations create at most $3C = C'$ foolish splits. Thus

$$f \leq C'.$$

Combining the above inequalities shows

$$s_4 - C' \leq \frac{g}{2} + f + r - C' \leq \frac{g}{2} + \frac{5}{6}(f + r) + \frac{C'}{6} + \frac{g}{3} - C'$$

$$= \frac{5}{6}(g + r + f - C') \leq \frac{5}{6}(s_1 - C').$$

This proves (9).

For convenience set $s_i = 0$ if i is larger than the number of iterations, so i ranges over all integers ≥ 1.

LEMMA 6.6. (a) There are $O(\log n_s)$ iterations of GENERAL-SPLIT.

(b) $\sum_i s_i = O(n_s)$.

Proof. We can apply (9) to an arbitrary graph H_1. Hence for any index i,

$$s_{i+3} - C' \leq \frac{5}{6}(s_i - C').$$

This shows there are $O(\log n_s)$ iterations where s has more than C' neighbors. It also implies the sum of all values s_i that are larger than C' is $O(n_s)$.

To complete the proof (of both parts) it suffices to show that $s_i \leq C'$ in $O(1)$ iterations i. To do this we strengthen our proof of termination of GENERAL-SPLIT. We claim that every iteration of GENERAL-SPLIT either decreases s_i or increases the number of foolish splits that exist in H. This claim completes the proof; it implies that when $s_i \leq C'$, s_i decreases after at most $2C'$ iterations (since a vertex is in at most two foolish splits).

To prove the claim, consider an iteration that does not decrease s_i. A split su, sv that is foolish at the start of the iteration is foolish in the next iteration. In proof, since the contraction step does not merge any neighbors of s, the next graph has $\overline{u} \neq \overline{v}$, $d(\overline{u}) = d(\overline{v}) = k$, and $d(\overline{u}, \overline{v}) \geq h$. The last inequality actually holds with equality, $d(\overline{u}, \overline{v}) = h$. (Otherwise $2k = d(\overline{u}) + d(\overline{v}) = d(\overline{u} \cup \overline{v}) + 2d(\overline{u}, \overline{v}) \geq k + 2(h+1) = 2k+1$, a contradiction.) So $s\overline{u}, s\overline{v}$ is a foolish split.

Now consider the first split sx, sy output by G-SPLIT. Recall this split is not foolish, and our proof of termination of GENERAL-SPLIT shows HOOK-UP returns a set $Y \in \mathcal{Y}$ containing x and y. In the Contraction Step Y has a k-bisection (since the Contraction Step does not merge neighbors of s). Thus the contractions make $s\overline{x}, s\overline{y}$ a new foolish split. □

As with EVEN-SPLIT, the total number of distinct splits executed in Step 2 in all iterations is $O(n_s)$, for both G and H. This follows from the same argument as EVEN-SPLIT, using Lemma 6.6(b) in the analysis of graph H.

Now we show the total time for GENERAL-SPLIT is $O(n(m + n \log n) \log n_s)$. The previous paragraph shows each graph H_i has $O(m+n_s) = O(m)$ edges. Each call to HOOK-UP takes time $O(n(m+n \log n))$. Testing each graph $H[Y]$ for a k-bisection involves running the edge connectivity algorithm of [16]. It uses time $O(n(m+n \log n))$ on a graph of n vertices and m edges. In one Contraction Step, the total number of vertices in all graphs $H[Y]$ is at most n and each $H[Y]$ has at most m edges. This implies the total time for one Contraction Step is $O(n(m+n \log n))$. (A tighter analysis shows all k-bisection tests in the entire algorithm use total time $O(n(m + n \log n))$.) The remaining processing in GENERAL-SPLIT takes time $O(m)$ in each iteration. Lemma 6.6(a) shows there are $O(\log n_s)$ iterations, so the desired time bound follows.

The same analysis shows that the time on unit capacity graphs is $O(nm \log n_s)$. We use the fact that the edge connectivity algorithm of [16] runs in time $O(nm)$ on these graphs. In summary we have proved the following refinement of [18].

THEOREM 6.7. *Procedure GENERAL-SPLIT finds a complete admissible splitting at a vertex s in time $O(n(m + n \log n) \log n_s)$ and space $O(m)$. The splitting contains $O(n_s)$ distinct splits. The time on unit capacity graphs is $O(nm \log n_s)$.* □

6.4. Complete splittings respecting partitions and degree sequences. The main task in adapting the algorithm of section 6.3 to allowed splitting is modifying G-SPLIT to create only splits that join two different partition classes. This section solves this problem.

Before discussing the problem recall that the *degree sequence* of a graph consists of the integers $d(x)$, where x ranges over every vertex. Hakimi [10] proved that given integers d_i, $i = 1, \dots, p$ are the degree sequence of a multigraph with no self-loops if and only if $\sum_i d_i$ is even and $d_1 \le \sum_{i>1} d_i$, where the indexing is chosen so $d_1 = \max\{d_i : 1 \le i \le p\}$.

Recall that E-SPLIT finds a complete splitting at s, with no split joining a vertex to itself. In the given (multi)graph let s have neighbors $x_i, i = 1, \dots, p$, and let d_i denote $d(s, x_i)$. The splitting found by E-SPLIT exists precisely when d_i is the degree sequence of a multigraph with no self-loops. Now it is clear that our algorithm E-SPLIT gives an alternate proof of Hakimi's theorem. (The necessity of Hakimi's condition is clear.) The algorithms of this section for complete splitting give other theorems on degree sequences, which we state with the algorithm.

We now state the problem solved in this section more precisely. In the graph G that is given for allowed splitting, let the partition be $\mathcal{P} = \{P_i : 1 \le i \le r\}$. Recall that our routines E-SPLIT and G-SPLIT are executed on a contraction H of G. For each neighbor x of s in H and each partition class P_i, $1 \le i \le r$, let $d_i(s, x)$ denote the total number of edges joining s to a vertex v of G that is in both vertex x and P_i. Thus $d(s, x) = \sum_i d_i(s, x)$. The task of E-SPLIT, when we compute allowed splittings, is to output a complete splitting that can be written as splits sx_i, sy_j, where x and y are distinct vertices of H, i and j are indices of distinct partition classes, and the total multiplicity of all splits involving x_i is $d_i(s, x)$. (Note that in the initial iteration when $H = G$, a given vertex x has $d_i(s, x) > 0$ for at most one index i.) The task of G-SPLIT for allowed splittings is the same as E-SPLIT with the additional requirement of not creating too many foolish splits (as in Lemma 6.4).

6.4.1. Even target connectivity k. We begin with the algorithm used when the target k is even and the partition has exactly two classes. This routine is of special interest because it is used for bipartite connectivity augmentation. Also, it is a subroutine of our algorithm for general partitions.

Let us restate the problem to be solved in a convenient notation. We are given nonnegative integers d_{i1}, d_{i2}, $i = 1, \dots, p$, such that

$$\sum_i d_{i1} = \sum_i d_{i2}. \tag{10}$$

(Note that in our splitting algorithm the index i corresponds to vertex x_i, d_{i1} and d_{i2} correspond to $d_1(s, x_i)$ and $d_2(s, x_i)$, respectively.) The task of the problem is to output a *complete pairing*, defined as a collection of ordered pairs of indices ii' where $i \neq i'$, with corresponding nonnegative multiplicities $\gamma_{ii'}$, such that $d_{i1} = \sum_{i'} \gamma_{ii'}$ and $d_{i'} = \sum_i \gamma_{ii'}$. Choose indices so that $d_{11} + d_{12} = \max\{d_{i1} + d_{i2} : 1 \le i \le p\}$. We will show that a complete pairing exists if

$$d_{11} + d_{12} \le \sum_{i>1} d_{i1} + d_{i2}. \tag{11}$$

First note the following interpretation in terms of degree sequences. Recall that a party graph is a complete bipartite graph with a perfect matching deleted. (Every person at the party talks to every member of the opposite sex except his or her spouse.) We generalize this slightly. For index i ranging from 1 to p, say that d_{i1}, d_{i2} is the *degree sequence of a party graph* if there is a bipartite multigraph with vertices x_{i1} on one side, x_{i2} on the other side, no edge $x_{i1}x_{i2}$, and $d(x_{ij}) = d_{ij}$ for $j = 1, 2$. We prove d_{i1}, d_{i2} is the degree sequence of a party graph if and only if (10) and (11) hold, where the indexing is as described above. The necessity of these conditions is clear.

Our algorithm for two partition classes and even k is the following procedure BE-SPLIT. It maintains (10) and (11) throughout its execution.

Procedure BE-SPLIT.

Step 1. While (11) holds with strict inequality, repeat the following Step 1(a).

Step 1(a). Choose distinct indices $i, i' > 1$ with $d_{i1}, d_{i'2} > 0$. Output pair ii' with multiplicity

$$\gamma = \min\left\{d_{i1}, d_{i'2}, \frac{\sum_{i>1}(d_{i1} + d_{i2}) - (d_{11} + d_{12})}{2}\right\}.$$

Decrease d_{i1} and $d_{i'2}$ by γ.

Step 2. (At this point (11) holds with equality.) Output pairs $1i$ with multiplicity d_{i2} and $i1$ with multiplicity d_{i1}, for all indices $i > 1$ that have positive d_{i2} or d_{i1}, respectively.

We show that BE-SPLIT outputs a complete pairing. Consider Step 1. First we show the indices in Step 1(a) exist.

We claim some index $i > 1$ has d_{i1} positive. In proof (10) shows $\sum_{i>1} d_{i2} = \sum_i d_{i1} - d_{12}$. Substituting in (11) with strict inequality gives

$$d_{11} + d_{12} < 2\sum_{i>1} d_{i1} + d_{11} - d_{12},$$

which implies the claim. Similarly some $i' > 1$ has $d_{i'2}$ positive. If these indices i and i' are distinct they form the desired pair for Step 1(a). Suppose they are identical, i.e., some $i > 1$ has $d_{i1}, d_{i2} > 0$. An index $i' > 1$ distinct from i has $d_{i'1} + d_{i'2}$ positive, since (11) holds with strict inequality and $d_{i1} + d_{i2} \le d_{11} + d_{12}$ ($d_{11} + d_{12} = \max\{d_{i1} + d_{i2} : i \ge 1\}$ throughout the algorithm). Thus either ii' or $i'i$ forms the desired pair.

Next observe that Step 1 eventually halts. This follows because the two sides of (11) have the same parity (by (10)). Each pair output maintains this relation. Thus eventually (11) holds with equality and Step 1 halts.

Finally we show Step 2 works correctly, i.e., the pairs that are output exist and account for all remaining values d_{i1}, d_{i2}. When we have equality in (11) the inequality displayed above becomes an equality and implies $d_{12} = \sum_{i>1} d_{i1}$. Similarly, $d_{11} = \sum_{i>1} d_{i2}$. These two relations show the pairs in Step 2 exist and exhaust all remaining values d_{i1}, d_{i2}, i.e., a complete pairing has been output.

Next we analyze the efficiency of BE-SPLIT. It outputs $O(p)$ distinct pairs. The argument is the same as for E-SPLIT: excluding the last pair output in Step 1, every pair decreases some d_{i1} or d_{i2} to zero.

BE-SPLIT can be implemented to use total time $O(p)$. In Step 1 we maintain a list of all indices $i > 1$ having positive d_{i1}, and we maintain a similar list for d_{i2}. An execution of Step 1(a) chooses the first index from both lists; if they are equal it also chooses the second index from one of the lists.

This concludes the discussion of BE-SPLIT. (We have also proved the theorem on party graph degree sequences.) We turn to the problem for general partitions (and even target k). As before we first restate the problem in a convenient notation. We are given nonnegative integers d_{ij} for $i = 1, \ldots, p$ and $j = 1, \ldots, q$. In our splitting algorithm i corresponds to vertex x_i, j corresponds to partition class j, and d_{ij} corresponds to $d_j(s, x_i)$. The given values have these properties for an integer D:

$$\sum_{i,j} d_{ij} = 2D;$$

$$\sum_{j} d_{ij} \le D \quad \text{for} \quad i = 1, \ldots, p;$$

$$\sum_{i} d_{ij} \le D \quad \text{for} \quad j = 1, \ldots, q.$$

(Note $\sum_{i,j} d_{ij}$ equals $\sum_{i=1}^{p} \sum_{j=1}^{q} d_{ij}$.) The task of the problem is to output a *complete pairing*, defined as a collection of pairs of ordered pairs $ij, i'j'$, where $i \ne i'$ and $j \ne j'$, with corresponding nonnegative multiplicities $\gamma_{ij,i'j'} = \gamma_{i'j',ij}$, such that for every $i = 1, \ldots, p$ and $j = 1, \ldots, q$, $d_{ij} = \sum_{i',j'} \gamma_{ij,i'j'}$.

In terms of degree sequences we prove the following. Let indices i and i' range over $1 \le i, i' \le p$, j and j' range over $1 \le j, j' \le q$. Say that d_{ij} is the *degree sequence of a p, q-party graph* if there is a multigraph with vertices x_{ij}, edges $x_{ij}x_{i'j'}$, where $i \ne i'$ and $j \ne j'$, and degrees $d(x_{ij}) = d_{ij}$. We prove d_{ij} is the degree sequence of a p, q-party graph if and only if the three conditions displayed above hold. The necessity of these conditions is clear.

Our algorithm incorporates an additional goal of limiting the number of pairs ii' that have pairs $ij, i'j'$ which are output. It is convenient to state the algorithm using the following notation. Choose indices so that $\max\{\sum_j d_{ij} : 1 \le i \le p\} \cup \{\sum_i d_{ij} : 1 \le j \le q\}$ equals either $\sum_j d_{1j}$ or $\sum_i d_{i1}$. In the first case define $\delta_i = 1$, $\delta_j = 0$. In the second case define $\delta_i = 0$, $\delta_j = 1$. Thus in both cases

$$\max \left\{ \sum_j d_{ij} : 1 \le i \le p \right\} \cup \left\{ \sum_i d_{ij} : 1 \le j \le q \right\} = \sum_{i=\delta_i \text{ or } j=\delta_j} d_{ij}.$$

(In the summation on the right the indices range over all pairs i, j such that either $i = \delta_i$ and $1 \le j \le q$ or $j = \delta_j$ and $1 \le i \le p$.) We assume that $d_{ij} = 0$ if i or j is zero.

Our algorithm for an arbitrary partition and even k is the following procedure, PE-SPLIT. It maintains the following relations throughout its execution.

$$\sum_{i,j} d_{ij} \text{ is even,} \tag{12}$$

$$\sum_{i=\delta_i \text{ or } j=\delta_j} d_{ij} \le \sum_{i\ne\delta_i, j\ne\delta_j} d_{ij}. \tag{13}$$

Notice that for the given values d_{ij} these conditions are equivalent to those assumed for the input.

Procedure PE-SPLIT.

Step 1. While (13) holds with strict inequality, repeat the following Steps 1(a) and 1(b).

Step 1(a). Choose distinct indices $i, i' > \delta_i$ such that there are distinct indices $j, j' > \delta_j$ with $d_{ij}, d_{i'j'} > 0$.

Step 1(b). While (13) holds with strict inequality and while there are distinct indices $j, j' > \delta_j$ such that $d_{ij}, d_{i'j'} > 0$, output the pair $ij, i'j'$ with multiplicity $\gamma = \min\{d_{ij}, d_{i'j'}, (\sum_{i\ne\delta_i, j\ne\delta_j} d_{ij} - \sum_{i=\delta_i \text{ or } j=\delta_j} d_{ij})/2\}$. Decrease d_{ij} and $d_{i'j'}$ by γ.

Step 2. (At this point (13) holds with equality.) Output the remaining pairs by using procedure BE-SPLIT on input values d'_{i1}, d'_{i2} defined as follows. If $\delta_i = 1$, then

$$d'_{j1} = d_{1j},\ d'_{j2} = \sum_{i>1} d_{ij} \quad \text{for} \quad 1 \le j \le q.$$

If $\delta_j = 1$, then

$$d'_{i1} = d_{i1},\ d'_{i2} = \sum_{j>1} d_{ij} \quad \text{for} \quad 1 \le i \le p.$$

In both cases the pairs of the form ii' output by BE-SPLIT correspond to pairs of the form $ij, i'j'$ in the obvious way.

We now show PE-SPLIT outputs a complete pairing. Consider Step 1. First we show the indices in Step 1(a) exist. For any index $i \ne \delta_i$,

$$\sum_{j\ne\delta_j} d_{ij} \le \sum_j d_{ij} \le \sum_{i=\delta_i \text{ or } j=\delta_j} d_{ij} < \sum_{i\ne\delta_i} \sum_{j\ne\delta_j} d_{ij}.$$

The second inequality follows by the relation defining δ_i and δ_j (which holds throughout the algorithm). The third inequality restates the fact that (13) holds with strict inequality.

The displayed relation implies that $\sum_{j\ne\delta_j} d_{ij}$ is nonzero for at least two indices $i \ne \delta_i$, say i, i'. These indices have corresponding indices j, j' that can be used in Step

1(a) unless there is an index $r \neq \delta_j$ with $d_{ir}, d_{i'r} > 0$. To handle this case note that $\sum_{i \neq \delta_i} d_{ij}$ is nonzero for at least two indices $j \neq \delta_j$, say j, j' (the proof is analogous to the previous proof for i, i'). Without loss of generality $j \neq r$. We can choose $i'' \neq \delta_i$ so that $d_{i''j} > 0$, and w.l.o.g. $i'' \neq i$. Hence the indices i, i'', r, j can be used in Step 1(a).

Next observe that Step 1 eventually halts. This follows because the two sides of (13) have the same parity (by (12)). Each pair output maintains this relation. Thus eventually (13) holds with equality and Step 1 halts.

To show Step 2 works correctly we need only check that the input conditions (10) and (11) for BE-SPLIT hold. Equality (10) follows from equality in (13). Observe that (11) is equivalent to $d_{j1} + d_{j2} \leq \frac{\sum_i d_{i1}+d_{i2}}{2}$ for every index j. If $\delta_i = 1$, then

$$cd'_{j1} + d'_{j2} = \sum_i d_{ij} \leq \sum_j d_{1j} = \frac{\sum_{i,j} d_{ij}}{2},$$

which gives the equivalent form of (11). The same argument applies to the case $\delta_j = 1$.

Next we show that PE-SPLIT outputs a small number of pairs. Specifically only $O(p)$ pairs ii' have indices jj' for which $\gamma_{ij,i'j'}$ is positive.

To prove this first consider two indices i, i' chosen in Step 1(a). Without loss of generality assume this is not the last execution of Step 1(a). This implies that when Step 1(b) halts one of the following conditions holds:

(i) $d_{ij} = 0$ *for every index* $j > \delta_j$, *or a similar condition holds for* i';

(ii) *there is an index* j *such that* $d_{ij'} = d_{i'j'} = 0$ *for every index* $j' \neq j, j' > \delta_j$.

Before Step 1(a) neither condition holds. Thus condition (i) holds at most once for a given index i. For (ii) note that at least one of the indices i, i'—say i—has more than one positive value d_{ij}, $j > \delta_j$, before Step 1(a), but only one positive value after Step 1(b). Thus condition (ii) holds at most once for a given index i. We conclude Step 1(a) is executed for $O(p)$ distinct pairs i, i'.

Next consider Step 2. If $\delta_i = 1$ then every pair $ij, i'j'$ output in Step 2 has $i = 1$. So there are $O(p)$ distinct pairs i, i' in Step 2. If $\delta_j = 1$, the input to BE-SPLIT consists of p values d'_{i1}, d'_{i2}. Thus BE-SPLIT outputs $O(p)$ distinct pairs ii'. Each such pair corresponds to pairs $i1, i'j$, $1 < j \leq q$, output by PE-SPLIT. Thus again Step 2 outputs $O(p)$ distinct pairs i, i'.

PE-SPLIT can be implemented to use total time $O(p+q+v)$, where v is the number of positive values d_{ij}. To implement Step 1 we maintain $q+1$ lists corresponding to these sets:

$$T = \{i : i > \delta_i, \text{ at least two indices } j > \delta_j \text{ have } \quad d_{ij} > 0\},$$
$$O_j = \{i : i > \delta_i, j \text{ is the unique index } > \delta_j \text{ with } d_{ij} > 0\}, \qquad \delta_j < j \leq q.$$

In addition each index i has a list of the indices j that have $d_{ij} > 0$. If $|T| \geq 2$, Step 1(a) chooses two indices i, i' from T. The lists for i and i' are used to generate the indices j, j' for Step 1(a). If $|T| \leq 1$, Step 1(a) chooses indices i and i' from two distinct lists (among T and the O_j) and generates indices j, j' in a similar way. The implementation of Step 2 is straightforward.

This concludes the discussion of PE-SPLIT. (We have also proved the theorem on p, q-party graph degree sequences.)

6.4.2. General targets. We adapt the algorithm PE-SPLIT to P-SPLIT, the routine for general partitions and arbitrary k. As before assume the given values

d_{ij}, $i = 1, \dots, p$, $j = 1, \dots, q$, satisfy (12) and (13). Procedure P-SPLIT outputs a complete pairing, which satisfies additional constraints coming from the graph H associated with the input values. Recall that index i corresponds to vertex x_i, with $d_{ij} = d_j(s, x_i)$. The pair $ij, i'j'$ corresponds to split $sx_i, sx_{i'}$. P-SPLIT must limit the number of foolish splits corresponding to its pairs; the precise sense of this requirement is given in Lemmas 6.8 and 6.9 below.

Recall that the indices of splits $sx_i, sx_{i'}$ are chosen in Step 1(a) of PE-SPLIT and Step 1(a) of BE-SPLIT. The latter is executed when BE-SPLIT is called from Step 2 of PE-SPLIT. The P-SPLIT routine uses a rule, called the Choice Step, to specify how the indices of splits are chosen. Note that if BE-SPLIT is called with $\delta_i = 1$ in PE-SPLIT, there is no need for such a rule, since in this case there is a unique complete splitting. Thus BE-SPLIT uses the Choice Step only when $\delta_j = 1$.

To use the same Choice Step for both Steps 1(a) we extend some notation from PE-SPLIT to BE-SPLIT. For BE-SPLIT define $\delta_i = 1$ and $\delta_j = 0$. Observe that these choices make the first sentence of Step 1(a) of BE-SPLIT identical to Step 1(a) of PE-SPLIT. (Although this convention allows us to treat the two routines uniformly, BE-SPLIT is not a special case of PE-SPLIT: the original definition of δ_i and δ_j for PE-SPLIT would lead us to choose $\delta_i = 0, \delta_j = 1$ because of (10).) Also define $q = 2$ for BE-SPLIT.

We need some more notation for the Choice Step. A *normal execution* of Step 1(a) is an execution of Step 1(a) of P-SPLIT or an execution of Step 1(a) of BE-SPLIT when BE-SPLIT is called with $\delta_j = 1$ in P-SPLIT. Observe that in a normal execution of Step 1(a) an index i in $1 \le i \le p$ corresponds to a vertex of H. (This is false for an abnormal execution.) For $\delta_j < j \le q$ define sets

$$I_j = \{i : \delta_i < i \le p,\ d_{ij} > 0\}.$$

Also for an index $\delta_i < i \le p$ let F_i denote the set of all indices $f > \delta_i$ with sx_i, sx_f a foolish split in the associated graph H. (This makes sense in a normal execution.)

The routine P-SPLIT is the same as PE-SPLIT except that normal executions of Step 1(a) choose indices i, i' by executing the following Choice Step.

Choice Step. Execute the case below that applies.

Case 1. There are two distinct indices $j, j' > \delta_j$ with $I_j \cap I_{j'} \neq \emptyset$.

Choose any $i \in I_j \cap I_{j'}$. Choose $i' \in \cup_t I_t - i$, with $i' \notin F_i$ if possible.

Case 2(a). There are exactly two nonempty sets I_j, disjoint and both of cardinality two.

Case 2(b). There are exactly three nonempty sets I_j, pairwise disjoint and all of cardinality one.

In both Cases 2(a) and 2(b), choose any index i in a set I_j such that $\cup_{t \neq j} I_t - F_i$ is nonempty, and choose an index i' in the latter set.

Case 3. No previous case applies.

Choose index $j > \delta_j$ so that I_j is nonempty and $|I_j|$ is minimum. Choose any $i \in I_j$. Choose an index $i' \in \cup_{t \neq j} I_t$, with $i' \notin F_i$ if possible.

When the target k is even there are no foolish splits, so the Choice Step does not substantively change PE-SPLIT. Also note that Case 2(b) never holds in BE-SPLIT. The analysis of PE-SPLIT shows P-SPLIT outputs a complete pairing if it is well defined, i.e., the index i always exists in Case 2. The following two lemmas state properties of the pairs output by P-SPLIT in terms of the corresponding splits on H, the graph corresponding to the input for P-SPLIT. As in section 6.3 H satisfies (1). Let H have vertex set $V_H + s$.

LEMMA 6.8.

(a) *P-SPLIT is well defined.*

(b) *P-SPLIT outputs at most $C = 3$ foolish splits.*

Proof. Consider a normal execution of Step 1(a). We discuss each case of the Choice Step.

Case 1. There are two distinct indices $j, j' > \delta_j$ with $I_j \cap I_{j'} \neq \emptyset$.

For any $i \in I_j \cap I_{j'}$, $d(s, x_i) \geq d_{ij} + d_{ij'} \geq 2$. Hence vertex x_i is in at most one foolish split and $|F_i| \leq 1$ (recall the discussion preceding the Choice Step in section 6.3). Thus Case 1 does not create a foolish split when $|\cup_t I_t| \geq 3$. We conclude that Case 1 creates a foolish split only when $|\cup_t I_t| \leq 2$.

Case 2(a). There are exactly two nonempty sets I_j, disjoint and both of cardinality two.

Case 2(b). There are exactly three nonempty sets I_j, pairwise disjoint and all of cardinality one.

We must show some index i in a set I_j has $\cup_{t \neq j} I_t - F_i$ nonempty. Suppose on the contrary that for every $j > \delta_j$ and $i \in I_j$, $\cup_{t \neq j} I_t \subseteq F_i$. Thus $d(x_i) = k$ and $d(x_i, x_r) = h$ for every $r \in \cup_{t \neq j} I_t$. The hypotheses of Cases 2(a) and 2(b) both imply there are two such indices r. Thus $d(s, x_i) = 1$.

Let $X = V_H - \{x_i : i \in \cup_j I_j\}$. Observe that X is nonempty. If $\delta_i = 1$, then $x_1 \in X$. If $\delta_j = 1$, then there are vertices y with $d_1(s, y) > 0$. Such a y does not correspond to an index in any I_j since, as noted above, any $i \in I_j$ has $d(s, x_i) = 1$, so $d(s, x_i) = d_j(s, x_i)$.

The equations of the first paragraph of this case imply $d(X, V_H - X) = 0$. We claim $d(s, X) \leq 2$. The claim implies $d(X) = d(s, X) \leq 2$. Using $k \geq 3$ this violates (1), giving the desired contradiction.

To prove the claim first suppose $\delta_i = 1$. In this case $d(s, X) = d(s, x_1)$. For Step 1(a) of P-SPLIT strict inequality holds in (13), so $d(s, x_1) \leq 4 - 2 = 2$. This inequality is also valid for Step 1(a) of BE-SPLIT since strict inequality holds in (11). Next suppose $\delta_j = 1$. In this case $d(s, X) = d_1(s, X)$. Again since strict inequality holds in (13), $d_1(s, X) \leq d_1(s, V_H) \leq 4 - 2 = 2$.

In summary we have proved the claim, thus showing the desired index i of the Choice Step exists. This gives part (a) of the lemma. We have also shown this case does not create a foolish split.

Case 3. No previous case applies.

Suppose $|\bigcup_j I_j| \geq 4$. We observe that selecting j as in the Choice Step gives $|\bigcup_{s \neq j} I_s| \geq 3$. In proof first note the sets I_j are pairwise disjoint, by Case 1. The observation follows if $|I_j| \geq 3$, since there is at least one set I_t. If $|I_j| = 2$ the observation follows from the definition of Case 2(a). If $|I_j| = 1$ the observation follows from the supposition $|\bigcup_j I_j| \geq 4$.

The observation implies that any $i \in I_j$ has an index $i' \in \bigcup_{t \neq j} I_t$ not creating a foolish split. We conclude that Case 3 creates a foolish split only when $|\bigcup_j I_j| \leq 3$.

We have shown the Choice Step does not create a foolish split unless $|\bigcup_j I_j| \leq 3$. This implies Step 1(a) of P-SPLIT creates at most one foolish split (recall the discussion preceding the Choice Step in section 6.3). If $\delta_i = 1$ in P-SPLIT, then BE-SPLIT creates at most one foolish split (since every split involves vertex 1). Suppose $\delta_j = 1$ in P-SPLIT. Then as above, Step 1(a) of BE-SPLIT creates at most one foolish split. In Step 2 of BE-SPLIT all the splits output involve a common vertex (the vertex indexed as 1 in BE-SPLIT). Thus Step 2 of BE-SPLIT creates at most one foolish split. In summary, at most three foolish splits are created, giving part (b) of the lemma. □

For the next lemma assume G and H satisfy the invariant of section 6.3. (We show this is the case in the next section.)

LEMMA 6.9. *If the first split output by P-SPLIT is foolish, then executing a maximal admissible subset of the splits output gives a new graph G containing a C_4-obstacle with $d(s) = 4$.*

Proof. The main case is when the first split is output in a normal execution of Step 1(a). We start by showing that if this is not the case, i.e., the first split is output in Step 1(a) of an abnormal execution of BE-SPLIT or Step 2 of BE-SPLIT, then the first split is not foolish. In fact we show BE-SPLIT finds a complete admissible splitting. To prove this observe that in both of these situations

$$(s, x_1) = \sum_{i>1} d(s, x_i).$$

This follows because if the first split is output in BE-SPLIT with $\delta_i = 1$ in P-SPLIT, equality holds in (13). Similarly, if the first split is output in Step 2 of BE-SPLIT with $\delta_j = 1$ in P-SPLIT, equality holds in (11). (In this case x_1 is the vertex indexed 1 in BE-SPLIT.) The displayed equation implies H has a unique complete splitting with no split joining a vertex to itself. Hence this splitting is found by BE-SPLIT. The splitting is admissible by the second remark at the end of section 6.1.

Now we turn to the main case, when the first split is output in a normal execution of Step 1(a). Fix the sets I_j to their values right before this first split is output. We claim the first split is foolish only when exactly two of these sets are nonempty, say I_j and $I_{j'}$, and for distinct indices $a, b, c > \delta_i$,

$$I_j = \{a\} \quad \text{and} \quad I_{j'} = \{b, c\}.$$

Cases 1–3 of Lemma 6.8 show the claim holds if $|\cup_j I_j| \geq 4$, or if $|\cup_j I_j| = 3$ and either two sets I_j have a nonempty intersection or the nonempty sets I_j are pairwise disjoint singletons. Since the indices of Step 1(a) exist, $|\cup_j I_j| \geq 2$. Hence the remaining case to prove the claim is $|\cup_j I_j| = 2$. Let $\cup_j I_j = \{i, i'\}$. We must show that the split $sx_i, sx_{i'}$ is not foolish. We will show it is admissible.

First observe that because the execution of Step 1(a) is normal, $d(s, \{x_i, x_{i'}\}) > d(s)/2$. (If $\delta_i = 1$, then $d(s, \{x_i, x_{i'}\})$ equals the right-hand side of (13) or (11). If $\delta_j = 1$, then $d(s, \{x_i, x_{i'}\})$ is at least the right-hand side of (13).) This implies any complete splitting with no split joining a vertex to itself contains the split $sx_i, sx_{i'}$. Hence this split is admissible.

We have proved the claim. To prove the lemma it remains only to treat the case of sets $I_j, I_{j'}$ displayed above.

Suppose for these sets the first split output by P-SPLIT is foolish. The Choice Step takes $i = a$, so $I_{j'} \subseteq F_a$. This implies $d(x_a) = d(x_b) = d(x_c) = k$, $d(x_a, x_b) = d(x_a, x_c) = h$, and $d(s, x_a) = 1$.

The first split is output in an execution of BE-SPLIT. This follows because strict inequality in (13) is impossible as it implies this contradiction:

$$d_{j'}(s, \{x_b, x_c\}) \leq d_{j'}(s, V_H) \leq d_j(s, x_a) + d_{j'}(s, \{x_b, x_c\}) - 2 = d_{j'}(s, \{x_b, x_c\}) - 1.$$

We conclude that the split is output in BE-SPLIT, so $\delta_i = 1$ and (11) holds with strict inequality.

Define $A = V_H - \{x_a, x_b, x_c\}$. This set is nonempty since it contains x_1.

The criticality of x_b implies

$$d(x_b, A) \leq h + 1 - d(s, x_b).$$

Similarly, $d(x_c, A) \leq h + 1 - d(s, x_c)$ and $d(x_a, A) = 0$. Since strict inequality holds in (11),

$$d(s, x_1) \leq d(s, x_b) + d(s, x_c) + d(s, x_a) - 2 = d(s, x_b) + d(s, x_c) - 1.$$

These inequalities imply $d(A) = d(A, x_b) + d(A, x_c) + d(A, x_a) + d(s, x_1) \leq 2h + 1 = k$. Hence by (1), equality holds throughout. Equality in the first displayed inequality implies $d(x_b, x_c) = 0$. Equality in the second displayed inequality implies $d(s, x_1) + d(s, x_a) = d(s, x_b) + d(s, x_c) = d(s)/2$. By (10), $d(s, x_1) = d_j(s, x_1)$. Examining the definition of d'_{ij} (for $\delta_j = 1$) in Step 2 of P-SPLIT shows that partition 1 in BE-SPLIT is the same as the original partition 1 in H. Let X_1, X_a, X_b, X_c denote the set of all vertices of G contained in x_1, x_a, x_b, x_c, respectively. In graph G either $d(s, X_1 \cup X_a) = d_1(s, X_1 \cup X_a)$ or $d(s, X_b \cup X_c) = d_1(s, X_b \cup X_c)$. Now we have deduced that A, X_a, X_b, X_c is a C_4-obstacle in G.

Without loss of generality, Step 1(a) outputs the (foolish) split sx_a, sx_b with multiplicity one. This gives equality in (11). The remaining splits are sx_1, sx_c with multiplicity $d(s, x_c)$ and sx_1, sx_b with multiplicity $d(s, x_b) - 1$. It is easy to see that a maximal admissible splitting executes all these splits except the split sx_a, sx_b and one copy of the split sx_1, sx_c (use Proposition 6.1). This gives the lemma. ∎

Since P-SPLIT is a special case of PE-SPLIT, it has the same bound on the number of pairs output; specifically, $O(p)$ pairs ii' have indices jj' for which $\gamma_{ij,i'j'}$ is positive. It is easy to implement P-SPLIT in time $O(m + n)$. We use the same data structures as for BE-SPLIT.

6.5. Computing a near-complete allowed splitting. Lemma 5.1 shows that any graph has a splitting of allowed pairs that is either complete or gives a graph containing a C_4-obstacle with $d(s) = 4$. We call this a *near-complete allowed splitting*. This section adapts the algorithm of section 6.3 to find such a splitting.

We note that Nagamochi and Ibaraki [17] give an algorithm for this task when k is even. The algorithm achieves the same time bound as ours in the special case that all edges have unit capacity.

The new algorithm is a straightforward extension of GENERAL-SPLIT, but we state it below for convenience. Recall the notation introduced at the start of section 6.4 to represent partitions in contracted graphs: $d_i(s, x)$ for degrees and sx_i, sy_j for splits.

The procedure TRANSLATE is called with two disjoint sets $X, Y \subset V$ and integers i, j, γ. X and Y have been contracted to vertices in H, and the split sX_i, sY_j with multiplicity γ is admissible in H.

Procedure TRANSLATE(X, i, Y, j, γ)**.**
Repeat Step 1 until $\gamma = 0$.

Step 1. Choose a vertex x of G corresponding to X: If X is a vertex of G let $x = X$. Otherwise choose a vertex $x \in X$ that is a neighbor of s in partition class P_i. Similarly choose a vertex y of G corresponding to Y, in partition class P_j. Execute split sx, sy in G with multiplicity $\gamma_0 = \min\{d(s, x), d(s, y), \gamma\}$ (d is the degree function in G). Decrease γ by γ_0.

Procedure PARTITION-SPLIT is our allowed splitting algorithm. It executes a near-complete allowed splitting at s on the original graph G. It starts with H initialized to the given graph G.

Procedure PARTITION-SPLIT.

Repeat Steps 1–3 until either $d(s) = 0$ or $d(s) = 4$ and H consists of four vertices forming a C_4-obstacle in G.

Step 1. Use graph G to compute the degree values $d_i(s, x)$ for each vertex x of H. Call P-SPLIT (with these degree values $d_i(s, x)$) to find a complete splitting at s in H, with splits sx_i, sy_j.

Step 2. Call HOOK-UP to execute a maximal admissible subset of these splits in H and to return the corresponding family of dangerous sets $\mathcal{Y}$. For each split that was executed in H call TRANSLATE to execute it in G.

Step 3. In H do the following for each set $Y \in \mathcal{Y}$. If Y is critical, contract Y. Otherwise ($d(Y) = k + 1$) test if Y has a k-bisection. If it does not, contract Y. If it does (say, into sets A and B), contract A and contract B.

To show correctness first observe that PARTITION-SPLIT maintains the same invariant as GENERAL-SPLIT, by exactly the same argument. Also observe that P-SPLIT is called with degree values $d_i(s, x)$ that satisfy (12) and (13): (12) holds since $d(s)$ is always even; (13) holds in the first iteration of PARTITION-SPLIT because of the initial choice of edges sx. (This is done by the algorithm of Lemma 4.1, which we implement in section 6.6.) Now we show (13) holds in subsequent iterations. For $\delta_i = 1$ (13) holds because every neighbor of s is in a dangerous set. Suppose $\delta_j = 1$. It suffices to show that each iteration preserves the fact that any partition j has $\sum_i d_{ij} \le d(s)/2$. This holds because each split involves two distinct partition classes, so each unexecuted split increases $\sum_i d_{ij}$ for two different partitions j.

It remains only to show that the main loop terminates the algorithm correctly, which we now do. Call an iteration of PARTITION-SPLIT *normal* if it performs at least one normal execution of Step 1(a), i.e., P-SPLIT starts with either strict inequality in (13) or equality in (13), $\delta_j = 1$ and strict inequality in (11). We will show that an execution of PARTITION-SPLIT begins with a finite number of normal iterations starting with inequality in (13), followed by a finite number of normal iterations starting with equality in (13) and inequality in (11), followed by at most one abnormal iteration.

First observe that once (13) holds with equality, it will hold with equality throughout the rest of the algorithm. This follows since equality means $d(s, x_1) = \sum_{i>1} d(s, x_i)$ or $d_1(s, V) = \sum_{j>1} d_j(s, V)$, and both of these relations are preserved by all subsequent splits and contractions.

Next observe that an iteration starting with equality in both (13) and (11) is abnormal. Furthermore, an abnormal iteration is the last iteration of PARTITION-SPLIT. This was shown in the first paragraph of the proof of Lemma 6.9.

It remains only to show that there is a finite number of normal iterations. The finiteness property holds because of a property similar to GENERAL-SPLIT: for indices $j \ge 0$ define

$$N_j = \{x : \text{vertex } x \text{ of } H \text{ has } d_i(s, x) > 0 \text{ for at least } j \text{ partition classes } i\}.$$

Thus N_0 is V_H and N_1 is the set of neighbors of s in H. Also $N_j = \emptyset$ for $j \ge 3$ in BE-SPLIT, i.e., when (13) holds with equality. The property is that, with two exceptions, every normal iteration ending with strict inequality in (13) or (11) decreases $|N_j|$ for some j in $0 \le j \le 3$. The two exceptions are the last iteration and an iteration that starts with strict inequality in (13) and ends with equality in (13). Note that the sets N_j in this termination property are defined using the given partition in iterations that start with inequality in (13) and the partition for BE-SPLIT in iterations that start with equality in (13). The exceptional iteration just mentioned switches the definition of sets N_j, since subsequent iterations execute BE-SPLIT. Note that this switch does not increase any $|N_j|$, since it just merges partition classes.

Our termination property implies there is a finite number of normal iterations: once a given vertex leaves N_j, $j \geq 1$, it never reenters N_j. Hence a sequence of normal iterations that preserves the number of vertices of H has finite length and leads to a decrease of $|N_0|$. Now we prove the termination property. Let the first split output by P-SPLIT be sx, sy with multiplicity γ. We begin by assuming this split is not foolish.

Suppose HOOK-UP splits off γ pairs of edges sx, sy. If sx, sy is the last split output in Step 1 of P-SPLIT or BE-SPLIT, then after executing this split equality holds in (13) or (11), respectively. So assume sx, sy is not the last split output in Step 1. Assume N_0 does not change in this iteration (else we are done). In Step 1 assume vertices x and y correspond to indices i and i', respectively. Recall the conditions (i) and (ii) that hold when Step 1(b) of P-SPLIT halts (see the end of section 6.4.1). These imply that (without loss of generality) one of these conditions holds when Step 1(b) of P-SPLIT halts: (i) $d_j(s,x) = 0$ for every index $j > \delta_j$; (ii) $d_j(s,x) > 0$ for more than one index $j > \delta_j$ before Step 1(a) but for only one such index after Step 1(b). It is easy to see that one of these conditions also holds after Step 1(a) of BE-SPLIT. We conclude that the iteration decreases $|N_r|$ for some index r, $1 \leq r \leq 3$. If (i) holds, then $r = 1$ if $\delta_j = 0$ and $r = 1$ or 2 if $\delta_j = 1$. If (ii) holds, then $r = 2$ if $\delta_j = 0$ and $r = 2$ or 3 if $\delta_j = 1$. (The assumption that N_0 does not change guarantees that at the end of the iteration x is still a vertex of H.)

On the other hand suppose HOOK-UP splits off fewer than γ pairs of edges sx, sy. HOOK-UP returns a set $Y \in \mathcal{Y}$ containing both x and y, and $Y \neq \{x, y\}$ since the split is not foolish. Hence the Contraction Step decreases $|N_0|$ when it processes set Y.

Now we consider the case that the first split sx, sy is foolish. We will show this is the last iteration of PARTITION-SPLIT. Lemma 6.9 implies that after HOOK-UP, graph G has $d(s) = 4$ and contains a C_4-obstacle. Let the obstacle consist of sets A_1, A_2, B_1, B_2 with a_1, a_2, b_1, b_2 their neighbors of s.

Without loss of generality the two splits that are not executed by HOOK-UP are sa_i, sb_i for $i = 1, 2$. HOOK-UP returns a dangerous set D_i containing a_i and b_i, for $i = 1, 2$. These sets are distinct, by Lemma 2.8, so they are disjoint.

Note the following general fact: If X is a critical set and Y is dangerous, with $X \cap Y$ containing a neighbor of s, then $X \subseteq Y$ or $Y \subseteq X$. (In proof suppose the contrary, i.e., $Y - X$ and $X - Y$ are nonempty. Then $k + (k+1) \geq d(X) + d(Y) \geq d(X - Y) + d(Y - X) + 2d(s, X \cap Y) \geq k + k + 2$, a contradiction.)

The fact implies $A_i \cup B_i \subseteq D_i$ for $i = 1, 2$. Disjointness of the D_i implies $D_i = A_i \cup B_i$. This implies $d(D_i) = k + 1$ (Lemma 2.4). Thus A_i and B_i form the unique k-bisection of D_i. The Contraction Step makes each of the four obstacle sets A_i, B_i, $i = 1, 2$ into a vertex. Hence the termination condition of the main loop now detects a C_4-obstacle on four vertices. We have thus shown the termination property for PARTITION-SPLIT.

The efficiency analysis requires a stronger termination property, which we now prove. Define

$$\Phi = 2|N_1| + |N_2| + |N_3|.$$

The property states that Φ never increases, and every normal iteration except possibly two either decreases Φ or increases the number of foolish splits that exist in H. The two exceptional iterations are the last iteration and an iteration that starts with strict inequality in (13) and ends with equality in (13).

To prove this property we analyze how Φ changes by computing first the change in Step 2 and then the change in Step 3. (First recall that Φ does not increase when we switch definitions of the N_j.) Step 2 can remove a vertex from a set N_j when it executes a split. Thus Step 2 can decrease Φ but it never increases Φ. To analyze *Step* 3 consider the effect of a contraction on Φ: Let vertices x_1 and x_2 be merged to form vertex X. Let $d_i(s, x_r) > 0$ for exactly j_r partition classes i, $r = 1, 2$, with $j_1 \geq j_2$. Then $d_i(s, X) > 0$ for between j_1 and $j_1 + j_2$ partition classes. Hence this operation does not change Φ if $j_2 = 0$ and it decreases Φ if $j_2 > 0$. We have now shown that an iteration never increases Φ.

The proof of the first termination property shows that with three exceptions the first split causes Φ to decrease, by either decreasing some $|N_j|$, $1 \leq j \leq 3$ or contracting a set that contains at least two neighbors of s. The three exceptions are the two exceptional iterations mentioned above, and when the first split creates a dangerous set with a k-bisection, whose two contractions do not change Φ. Now consider a normal iteration that is not one of the two exceptional iterations and does not change Φ. As in the proof of Lemma 6.6, every split that is foolish at the start of the iteration is foolish at the end of the iteration, and furthermore the first split causes a contraction creating a new foolish split. This proves the stonger termination property.

The efficiency analysis for PARTITION-SPLIT is essentially identical to previous arguments. PARTITION-SPLIT has $O(\log n)$ iterations, by the same derivation as for GENERAL-SPLIT; Lemma 6.5 is proved exactly the same way. This holds similarly for inequality (9). The proof of Lemma 6.6 incorporates the stronger termination property just proved. (This property implies that $s_i \leq C'$ in $O(1)$ iterations, since $s_i \leq C'$ implies $\Phi \leq 4C'$.)

The total number of distinct splits executed in Step 2 in all iterations is $O(n)$, for both G and H. The proof is the same argument as GENERAL-SPLIT. For the analysis of graph H we use the fact that each execution of P-SPLIT gives a splitting with $O(s_i)$ distinct splits sx, sy.

The time bound for PARTITION-SPLIT is $O(n(m + n \log n) \log n)$. (The time is $O(nm \log n)$ on unit capacity graphs.) The argument is the same as GENERAL-SPLIT, using the above facts.

6.6. Finding good neighbors. This section gives algorithms to compute the set of edges incident to s, i.e., the values $d(s, x)$. This problem occurs three times in our connectivity augmentation algorithm—in computing the initial graph, rearranging edges in a C_4-obstacle, and rearranging edges in a C_6-obstacle. We discuss each of these. Our approach is to simulate the procedures given in previous sections using an efficient algorithm for computing the initial edges of an admissible splitting.

Nagamochi, Nakamura, and Ibaraki [18] give an algorithm MINIMAL that starts with a graph $G = (V + s, E)$ satisfying (1) and deletes edges incident to s so (1) still holds but decreasing any value $d(s, x)$ destroys (1). MINIMAL is a modification of HOOK-UP and has the same time bound, $O(n(m + n \log n))$. The time is $O(nm)$ on unit capacity graphs.

We use a straightforward generalization of MINIMAL. In addition to G, this algorithm is given a value $\ell(s, x)$ for each $x \in V$, with $\ell(s, x) \leq d(s, x)$ in G. The graph that is returned satisfies (1) and has $\ell(s, x) \leq d(s, x)$ for every $x \in V$, and no value $d(s, x)$ can be decreased subject to these constraints. It is simple to modify the algorithm of [18] to handle these lower bounds. (Specifically the algorithm of [18] initializes the solution graph to G with each value $d(s, x)$ decreased to 0. It proceeds to add back a minimal set of edges of G incident to s that achieves (1). The only change for the modified algorithm is to decrease initially each $d(s,x)$ to $\ell(s,x)$.) The time bound for the modified algorithm is unchanged. We still call this modified algorithm MINIMAL.

6.6.1. Computing the initial graph. The initial edges incident to s are computed by Steps 1 and 2 of the procedure given in Lemma 4.1. (These steps are used for both even and odd k.) Step 1 computes values $d(s,x)$ that achieve (1) and are minimal. We do this using the original algorithm MINIMAL of [18]. The rest of Step 1 ensures that $d(s)$ is even, specifically $d(s) = 2\alpha$ for α as defined in section 4. This part is trivial to implement.

Step 2 applies if there is a partition class P_1 such that $d(s,P_1) > d(s,V-P_1)$. It decreases $d(s,P_1)$ and increases $d(s,V-P_1)$ by the same amount, preserving (1), until either the two quantities become equal or any further change destroys (1). The first case ends with $d(s,P_1) = d(s,V-P_1) = \alpha$; the second case ends with $d(s,P_1) > \alpha > d(s,V-P_1)$. Our implementation of Step 2 begins by deciding which of these two alternatives holds, as follows.

Call the degree function at the end of Step 1 d_1. Thus $d_1(s,P_1) > \alpha > d_1(s,V-P_1)$. Define values $d_2(s,x)$ and lower bounds $\ell(s,x)$ by

$$\begin{aligned} d_2(s,x) &= d_1(s,x), \ell(s,x) = 0 \quad &&\text{for} \quad x \in P_1;\\ d_2(s,x) &= \ell(s,x) = k &&\text{for} \quad x \in V - P_1. \end{aligned}$$

Execute MINIMAL starting with these values. Let d_2' denote the degree function in the graph that is returned. Note that d_2' corresponds to a minimal degree function satisfying (1) for sets $X \subseteq P_1$.

Clearly when the first case of Step 2 holds, we get $d_2'(s,P_1) \le \alpha$. It is not hard to see that when the second case of Step 2 holds we get $d_2'(s,P_1) > \alpha$. (Use the family of sets W_i defined in the discussion of Step 2(c).) We process the two cases as follows.

Case 1. $d_2'(s,P_1) \le \alpha$. As in Step 2 our goal is to construct a new degree function so $d(s,P_1) = d(s,V-P_1) = \alpha$. To do this first define values $d_2''(s,x)$ for $x \in P_1$ as follows. Initialize each $d_2''(s,x)$ to $d_2'(s,x)$. Then increase some values $d_2''(s,x)$, $x \in P_1$, maintaining $d_2''(s,x) \le d_1(s,x)$, until $d_2''(s,P_1) = \alpha$. Note that aside from these constraints, the vertices x whose values are increased and the amounts increased are arbitrary. It is possible to achieve the desired function d_2'' by the definition of d_2 and Case 1.

Now define values $d_3(s,x)$ and lower bounds $\ell(s,x)$ by

$$\begin{aligned} d_3(s,x) &= \ell(s,x) = d_2''(s,x) \quad &&\text{for} \quad x \in P_1;\\ d_3(s,x) &= k, \ \ell(s,x) = 0 &&\text{for} \quad x \in V - P_1. \end{aligned}$$

Execute MINIMAL starting with these values. Use the resulting graph as the output of Step 2.

To see that this construction is correct, observe that any $x \in P_1$ has $d_1(s,x) \ge d_2''(s,x)$. Suppose we execute Step 2 starting with the degree function d_1. In each execution of Step 2(b) we can choose u to be any vertex x satisfying $d_1(s,x) > d_2''(s,x)$, where the quantity on the left is the current value of d_1. This follows from our characterization of d_2' as a minimal degree function satisfying (1) for sets $X \subseteq P_1$. Hence we can execute Step 2 so it halts with a degree function d equal to d_2'' on vertices of P_1. This degree function has $d(s) = 2\alpha$. The existence of d implies that the above execution of MINIMAL returns a similar degree function d'_3, i.e., d'_3 equals d''_2 on P_1 and $d'_3(s) = 2\alpha$. Thus our algorithm behaves the same as Step 2.

Case 2. $d'_2(s, P_1) > \alpha$. Our characterization of d'_2 shows that Step 2 halts with a degree function satisfying $d(s, P_1) = d'_2(s, P_1) = d(s, V - P_1)$. Thus it suffices to construct a degree function satisfying this equation (and (1)). To do this define values $d_3(s, x)$ by

$$\begin{aligned} d_3(s,x) &= d'_2(s,x) \quad \text{for} \quad x \in P_1;\\ d_3(s,x) &= k \qquad\qquad \text{for} \quad x \in V - P_1. \end{aligned}$$

Execute MINIMAL (with no lower bounds) starting with these values. Let d'_3 be the resulting degree function. Note that $d'_3(s, V - P_1) < d'_3(s, P_1)$ (by Step 2). Now as in Step 2(c), increase values $d'_3(s, x)$ for $x \in V - P_1$ to make $d'_3(s, V - P_1) = d'_3(s, P_1)$.

We have shown this implementation of Step 2 is correct. It constructs the initial degree function in the same time bound as MINIMAL, $O(n(m + n \log n))$.

6.6.2. Rearranging obstacles. The basic step in rearranging obstacles operates on a given critical set $X \subset V$ and a given class P_i such that all edges from s to X go to vertices of P_i. The task is to check if this is forced—more precisely, if X has a subpartition $\mathcal{F}$ consisting of sets U, each contained in P_i, such that in $G - s$,

$$k - d(X) = \sum_{U \in \mathcal{F}} k - d(U). \tag{14}$$

If no such $\mathcal{F}$ exists, we may execute a second part of this basic step: the second part replaces one edge that goes from s to $X \cap P_i$ by an edge from s to $X \cap P_j$, $j \neq i$, still preserving (1).

We perform the basic step by computing a minimal degree function satisfying (1) for all subsets of $X \cap P_i$. Specifically, define values $d_1(s, x)$ and lower bounds $\ell(s, x)$ by

$$\begin{aligned} d_1(s,x) &= d(s,x),\ \ell(s,x) = 0 \quad \text{for} \quad x \in X \cap P_i;\\ d_1(s,x) &= \ell(s,x) = k \qquad\qquad \text{for} \quad x \in V - (X \cap P_i). \end{aligned}$$

Execute MINIMAL starting with these values. The resulting degree function d' is a minimal degree function satisfying (1) for subsets of $X \cap P_i$. By [6], $d'(s, X \cap P_i)$ equals the maximum possible value of the sum in (14). Note that the left-hand side of (14) (which is computed in $G - s$) equals $d(s, X)$ (since X is critical). We conclude that the condition of (14) is equivalent to the test $d(s, X) = d'(s, X \cap P_i)$.

If the test of (14) fails, then $d(s, X) > d'(s, X \cap P_i)$. In this case the second part of the basic step, replacing an edge from s to X, can be performed as follows. Choose any vertex $u \in X \cap P_i$ with $d'(s, u) < d(s, u)$. Consider an irreducible subpartition of X (as defined before Lemma 5.7) that satisfies (14) (here we do not restrict the sets U to be contained in P_i). Vertex u is in some set U of the partition. We cannot have $U \subseteq P_i$ since then $d'(U) < d(U) = k$. Choose any vertex $v \in U - P_i$. Lemma 2.12 shows that replacing edge su by sv preserves (1).

To perform the procedure just described, use the above vertex u to define values δ_x for $x \in X \cap P_i$ by $\delta_x = 1$ for $x = u$ and $\delta_x = 0$ otherwise. Define values $d_2(s, x)$ and lower bounds $\ell(s, x)$ by

$$\begin{aligned} d_2(s,x) &= \ell(s,x) = d(s,x) - \delta_x \qquad\qquad \text{for} \quad x \in X \cap P_i;\\ d_2(s,x) &= d(s,x) + 1,\ \ell(s,x) = d(s,x) \quad \text{for} \quad x \in V - (X \cap P_i). \end{aligned}$$

Execute MINIMAL starting with these values. The resulting degree function replaces edge su by an edge sv, preserving (1), as described above.

The above basic step is used to rearrange obstacles as follows. First consider a C_4-obstacle in $G = (V + s, E)$. Let the obstacle sets be A_1, A_2, B_1, B_2, where partition class P_1 contains all neighbors of s in $A_1 \cup A_2$. According to Lemma 5.7 either the obstacle sets form a C_4-configuration in the given graph, or rearranging up to two edges incident to s gives a graph with no C_4-obstacle. We implement the procedure of Lemma 5.7 as follows. Use the basic step to check if both sets A_1 and A_2 have the subpartition $\mathcal{F}$ of (14). If so, the obstacle sets form a C_4-configuration. If not, perform the second part of the basic step (to replace one edge) on one set not having such a subpartition. Repeat this procedure for sets B_1 and B_2 if some partition class P_2 contains all neighbors of s in $B_1 \cup B_2$.

A C_6-obstacle is handled similarly. We implement the procedure of Lemma 5.6. Use the basic step to check if each of the six obstacle sets has the subpartition $\mathcal{F}$ of (14) (this subpartition will contain just one set). If so, the obstacle sets form a C_6-configuration. If not, perform the second part of the basic step on one obstacle set not having such a subpartition.

6.7. The constrained connectivity augmentation algorithm. Now we combine the pieces of the previous sections to get our algorithm for partition-constrained edge-connectivity augmentation.

The algorithm uses a subroutine to find a complete allowed splitting in a graph having $d(s) = 6$, $d(s, P) \leq 3$ for every partition class P, and containing no obstacle. The splitting exists by Theorem 3.6. To find it try every possible complete splitting that respects the partition constraints. Test every resulting graph for k-edge-connectivity using the procedure of [16]. Return the desired complete allowed splitting.

The algorithm for partition-constrained edge-connectivity augmentation starts by executing the algorithm of section 6.6 to define the initial degree function $d(s, x)$.

Next execute PARTITION-SPLIT. If it finds a complete allowed splitting, then return. (This will be the case when the target connectivity k is even.) If the splitting is not complete, then PARTITION-SPLIT finds a C_4-obstacle with $d(s) = 4$. As in Theorem 5.8 consider the two cases.

Case 1. Every split that was executed joins vertices in different obstacle sets.

Test if the C_4-obstacle gives a C_4-configuration, using the procedure of section 6.6. If so, use the procedure of Lemma 5.1 to add three edges to G (the graph returned by PARTITION-SPLIT) to make it k-edge-connected. This is optimal, so return.

If there is no C_4-configuration, rearrange the C_4-obstacle using the procedure of section 6.6. Execute PARTITION-SPLIT on the new graph. If it finds a complete splitting, then return. If not, PARTITION-SPLIT finds a C_4-obstacle to which Case 2 applies, so execute that case.

Case 2. A split joining two vertices in the same obstacle set was executed.

Let sx, sy be such a split. Undo it to get a graph G' with $d(s) = 6$. If neither x nor y belongs to a partition P_i with $d(s, P_i)$ maximum in G' then G' contains has no obstacle. Find a complete allowed splitting using the subroutine described above.

In the opposite case change the edges of this split, as described in Theorem 3.6, Case 1, and execute the new split. This gives a graph with $d(s) = 4$. Execute PARTITION-SPLIT on this graph. If PARTITION-SPLIT finds a complete splitting, then return. (This will be the case when there are only two partition classes.) If not, PARTITION-SPLIT finds a second C_4-obstacle. Use the two C_4-obstacles to construct a C_6-obstacle, as described in Lemma 3.9.

Suppose the C_6-obstacle is not in the original graph. Change a preceding split to get a graph G'' with no obstacle and $d(s) = 6$, as in the proof of Lemma 3.5. Find a complete allowed splitting in G'' using the subroutine.

Suppose the C_6-obstacle is in the original graph. Test if the C_6-obstacle gives a C_6-configuration, using the procedure of section 6.6. If not, rearrange the C_6-obstacle using the procedure of section 6.6. This gives a graph G'' with no obstacle and $d(s) = 6$. Find a complete allowed splitting of G'' by using the subroutine.

The remaining case is when the original graph has a C_6-configuration. Using the last C_4-obstacle found, execute the procedure of Lemma 5.1 to add three more edges to make the graph k-edge-connected. This is optimal, so return.

In summary we have proved the following.

THEOREM 6.10. *The edge-connectivity augmentation problem with partition constraints can be solved for $k \geq 2$ in time $O(n(m + n \log n) \log n)$ and space $O(m)$. The time is $O(nm \log n)$ on unit capacity graphs.* □

7. An application in statics. In this section we show how one of our results (Corollary 5.10) solves an open question raised by Frank (and implicitly by Recski in his book [20, pp. 63–64]) and hence present a possible application in statics, a new field where connectivity augmentation algorithms may be useful. Corollary 5.10 gives a good characterization for the augmentation problem where the starting graph G is bipartite and the goal is to increase the edge-connectivity to k optimally by adding edges such that bipartiteness is preserved. (The polynomial algorithm which solves this problem is a simplified version of our main algorithm; see Theorem 6.10.)

One of the basic two-dimensional structures in statics is a *square-grid framework* consisting of horizontal and vertical *rods*, each of the same length and rigid, and *joints*, which connect the incident rods and which are rotatable. The rods collectively join all gridpoints in a rectangular region that are adjacent horizontally or vertically.

A square-grid framework can be deformed by rotating certain parts along certain joints. To prevent these deformations we can add rods (call them *extra rods*) to the framework diagonally into some of the squares. A framework (possibly containing extra rods) is *rigid* if, roughly speaking, it has no nontrivial deformations; that is, fixing the position of a rod of the grid to the plane, the positions of all other rods (and joints) are uniquely determined. (For a precise description and more details see [20, Section 2.6].)

How can we decide whether a framework (with extra rods) is rigid? The answer—which was found by Bolker and Crapo [2]—depends only on the *graph of the framework*. Given a square-grid framework S with m rows and l columns of squares, define the graph $G_S = (V, E)$ of S as follows. Let $V = \{v_1, \ldots, v_m, w_1, \ldots, w_l\}$ and connect v_i and w_j by an edge if there is an extra rod in the intersection of row i and column j. Observe that G_S is a bipartite graph and if at most one extra rod may be put into each square, G_S is simple as well.

THEOREM 7.1 (see [2]). *The square-grid framework S is rigid if and only if the corresponding graph G_S is connected.* □

One may want to brace a framework, by adding more extra rods, so it can survive the failure of $k' \geq 1$ extra rods; that is, in removing any set of at most k' extra rods, the framework remains rigid. We say that a square-grid framework S is *k-rigid* if S remains rigid by removing any set of at most $k' < k$ extra rods. This yields the following optimization problem: Given a square-grid framework S, find a smallest set F of new extra rods whose addition makes S k-rigid. In terms of graphs this corresponds to the augmentation problem we described at the beginning of this section and solved (without the additional simplicity assumption) in Corollary 5.10 and Theorem 6.10.

In the rest of this section we briefly discuss the case where simplicity must also be preserved. We show that if $k \leq 3$, then the (size of the) solution we obtain by our algorithm of section 6 is optimal in this case, too, by showing how the possible parallel edges in our solution can be substituted one by one by some other new edges maintaining k-edge-connectivity and achieving simplicity.

Let $G = (A, B, E)$ be a simple bipartite graph we want to make optimally k-edge-connected preserving bipartiteness and simplicity. We can assume that $|A|, |B| \geq k$ and $k \geq 2$. Let F be a solution produced by our algorithm of section 6 and suppose that $G' = (A, B, E \cup F)$ contains parallel edges.

First let $k = 2$ and let $f \in F$ be an edge which is parallel to some edge $e \in E \cup F$.

LEMMA 7.2. *There exists an edge $f' \notin E \cup F$ for which $G'' = (A, B, E \cup F - f + f')$ is bipartite and* 2*-edge-connected.*

Proof. By the optimality of F the edge f is critical with respect to 2-edge-connectivity in G'. Since e is parallel to f, we can observe that the only cut-edge in $G' - f$ is e. Let X and Y be the two connected components we obtain from $G' - f$ by removing e. It is enough to show that there exist two vertices u, v, one in X and one in Y, which are not adjacent in $G' - f$ and which belong to different classes of the bipartition (A, B). The existence of such a vertex pair can be checked easily (using $|A|, |B| \geq 2$). ◻

Consider the case $k = 3$. Using the same notation suppose that G' contains parallel edges and let $f \in F$ be an edge which is parallel to some edge $e \in E \cup F$.

LEMMA 7.3. *There exists an edge $f' \notin E \cup F$ for which $G'' = (A, B, E \cup F - f + f')$ is bipartite and* 3*-edge-connected.*

Proof. Let $f = xy$ and let $X \subset A \cup B$ ($Y \subset A \cup B$) be a minimal set containing x (y) with $d_{G'-f}(X) = 2$ ($d_{G'-f}(Y) = 2$, respectively). Minimality implies $X \cap Y = \emptyset$. The key observation is that $X \cup Y = A \cup B$ since $d_{G'}(X \cup Y) = d_{G'}(X) + d_{G'}(Y) - 2d_{G'}(X, Y) \leq 3 + 3 - 4 = 2$ (using that both e and f connect X and Y in G'). Thus the unique 2-edge-cut in $G' - f$ is induced by X (and $Y = V - X$). Therefore it is enough to show that there exist two vertices u, v, one in X and one in Y, which are not adjacent in $G' - f$ and which belong to different classes of the bipartition (A, B). The existence of such a vertex pair can be checked easily (using $|A|, |B| \geq 3$). ◻

The above lemmas cannot be extended to $k \geq 4$. (To see this take a bipartite 4-edge-connected simple graph $\bar{G} = (A, B, E)$ and extend it by two new vertices u, v, the edge uv, two distinct edges from u to A, and two distinct edges from v to B. The resulting bipartite graph is 3-edge-connected and its unique optimal augmentation with respect to $k = 4$ is a new copy of the edge uv.) Note also that for the special case $k = 1$ of the directed version of this bipartite augmentation problem a linear-time algorithm and a min-max formula were given in [8].

8. Remarks. Another type of "partition-constrained" augmentation problem is the following (let us call it Problem PC2).

INSTANCE $G = (V, E)$ *an undirected graph,* $\mathcal{P} = \{P_1, \ldots, P_r\}$ *a partition of* V, k, *and* M *integers.*

QUESTION *Does there exist a set F of edges such that $G' = (V, E \cup F)$ is k-edge-connected, $|F| \leq M$, and the endvertices of each edge of F lie within a class of $\mathcal{P}$?*

THEOREM 8.1. *Problem PC*2 *is NP-complete.*

Proof. The problem obviously belongs to the class NP. We reduce the Hamiltonian cycle problem (HCP) to PC2. Let $H = (W, E')$ be an instance of HCP. We can assume that H is 2-edge-connected. Let $W = \{w_1, \dots, w_n\}$.

Construct the following instance of PC2. For every vertex $w_i \in W$ we have a vertex v_i in G, and for every edge $w_i w_j \in E'$ let G contain two vertices u_{ij}^1 and u_{ij}^2, two parallel edges between u_{ij}^1 and v_i, and two parallel edges between u_{ij}^2 and v_j. Let $\mathcal{P}$ consist of the one-element classes $\{v_1\}, \dots, \{v_n\}$ and the classes $\{u_{ij}^1, u_{ij}^2\}$ of size two for every pair ij with $w_i w_j \in E'$. Let $M = n$ and $k = 2$. We claim that the problem PC2 has a solution with this instance if and only if H has a Hamiltonian cycle. The proof of this simple claim (and hence the theorem) is left to the reader. □

Note that the case where r is fixed (in particular if $r = 2$) in problem PC2 remains open.

Acknowledgments. The authors gratefully thank András Frank. He invited us to the Second Budapest Workshop on Network Design, where this work was begun, and he suggested the application of section 7. We thank Hiroshi Nagamochi for making a preprint of [18] available.

REFERENCES

[1] J. Bang-Jensen and T. Jordán, *Edge-connectivity augmentation preserving simplicity*, SIAM J. Disc. Math., 11 (1998), pp. 603–623.

[2] E.D. Bolker and H. Crapo, *How to brace a one-story building*, Environ. Plan. B, 4 (1977), pp. 125–152.

[3] G.-R. Cai and Y.-G. Sun, *The minimum augmentation of any graph to a k-edge-connected graph*, Networks, 19 (1989), pp. 151–172.

[4] E. Cheng and T. Jordán, *Successive edge-connectivity augmentation problems*, Math. Programming, Ser. B., to appear.

[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. Disc. Math., 5 (1992), pp. 25–53.

[7] A. Frank, *Connectivity augmentation problems in network design*, Math. 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] H.N. Gabow and T. Jordán, *How to make a square grid framework with cables rigid*, Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms, 1999, pp. 356–365.

[9] M. Grötschel, C.L. Monma, and M. Stoer, *Design of survivable networks*, Handbooks in Oper. Res. Management Sci. 7, Network Models, M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, eds., North–Holland, Amsterdam, 1994, pp. 617–672.

[10] S. L. Hakimi, *On realizability of a set of integers as degrees of the vertices of a linear graph*, J. SIAM, 10 (1962), pp. 496–506.

[11] T.-S. Hsu and M.-Y. Kao, *Optimal augmentation for bipartite componentwise biconnectivity in linear time*, Algorithms Comput., Lecture Notes in Comput. Sci. 1178, Springer-Verlag, Berlin, 1996.

[12] T. Jordán, *Two NP-Complete Augmentation Problems*, Preprint 8, Department of Math and Computer Science, Odense University, Odense, Denmark, 1997.

[13] G. Kant and H.L. Bodlaender, *Planar graph augmentation problems*, Algorithms and Data Structures, Lecture Notes in Comput. Sci. 519, Springer-Verlag, Berlin, 1991.

[14] L. Lovász, *Combinatorial Problems and Exercises*, North–Holland, Amsterdam, 1979.

[15] W. Mader, *A reduction method for edge-connectivity in graphs*, Ann. Discrete Math., 3 (1978), pp. 145–164.

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

[17] H. Nagamochi and T. Ibaraki, *Deterministic $\tilde{O}(nm)$ time edge-splitting algorithm in undirected graphs*, J. Combin. Optim., 1 (1997), pp. 5–46.
[18] H. Nagamochi, S. Nakamura, and T. Ibaraki, *A simple $\tilde{O}(nm)$ time edge-splitting algorithm in undirected graphs*, Tech. report 96013, Department of Applied Mathematics and Physics, Kyoto University, Kyoto, Japan, 1996.
[19] D. Naor, D. Gusfield, and Ch. Martel, *A fast algorithm for optimally increasing the edge connectivity*, SIAM J. Comput., 26 (1997), pp. 1139–1165.
[20] A. Recski, *Matroid Theory and its Applications in Electric Network Theory and in Statics*, Akadémiai Kiadó, Budapest, 1989.
[21] M. Stoer, *Design of Survivable Networks*, Ph.D. thesis, Lecture Notes in Math. 1531, Springer-Verlag, Berlin, 1992.
[22] T. Watanabe and A. Nakamura, *Edge-connectivity augmentation problems*, J. Comput. System Sci., 35 (1987), pp. 96–144.

SIAM J. DISCRETE MATH.
Vol. 12, No. 2, pp. 208–216

ON PERFECT MATCHINGS AND HAMILTON CYCLES IN SUMS OF RANDOM TREES*

ALAN FRIEZE†, MICHAŁ KAROŃSKI‡, AND LUBOŠ THOMA§

Abstract. We prove that the sum of two random trees possesses with high probability a perfect matching and the sum of five random trees possesses with high probability a Hamilton cycle.

Key words. sums of random trees, perfect matching, Hamilton cycle

AMS subject classification. 05C80

PII. S0895480196313790

1. Introduction.

In this paper we prove that an appropriately defined sum of two random trees possesses with high probability (w.h.p.) a perfect matching. Second, we show that the sum of five random trees possesses w.h.p. a Hamilton cycle.

We say that a sequence of events $\mathcal{E}_n$ (defined on a sequence of probabilistic spaces) holds *w.h.p.* if the probabilities of these events converge to 1 as $n \to \infty$.

For an integer n, we use $[n]$ to denote the set $\{1, \ldots, n\}$. A *random tree* on the set $V_n = [n]$ is a tree on this set chosen uniformly at random from the family of all trees on the set $[n]$.

DEFINITION 1.1 (sums). *Let k be a positive integer. For trees $T_1, \ldots, T_k$, all of them on the set $[n]$, we define their sum $\mathcal{ST}(T_1, \ldots, T_k)$ as the graph on the vertex set $[n]$ and edge set being the union of edge sets of the trees $T_1, \ldots, T_k$, where the parallel edges coalesce.*

Let f be a mapping from $[n] \to [n]$. Let $D(f)$ be its associated functional digraph, i.e., the graph with vertex set $[n]$ and edges $(i, f(i))$, $i \in [n]$. For a set $f_1, \ldots, f_k$ of such mappings we define their sum $\mathcal{SM}(f_1, \ldots, f_k)$ as the union of the digraphs $D(f_i)$, $1 \le i \le k$.

Let k be a positive integer. Consider k random trees $T_1, \ldots, T_k$ on $[n]$ chosen independently. We use the notation $\mathbf{ST}_n(k)$ for $\mathcal{ST}(T_1, \ldots, T_k)$.

A *random mapping* $f : [n] \to [n]$ is a mapping from the set $[n]$ to itself chosen uniformly at random from the family of all mappings $[n] \to [n]$. Similarly, as in the case for trees, we use $\mathbf{SM}_n(k)$ to denote the sum of k random mappings.

$\mathbf{SM}_n(k)$ is a well-studied model of a random graph. Frieze [4] showed that w.h.p. $\mathbf{SM}_n(2)$ has a perfect matching (see also Shamir and Upfal [11], who showed that w.h.p. $\mathbf{SM}_n(6)$ has a perfect matching). Cooper and Frieze [2] have shown that

*Received by the editors December 16, 1996; accepted for publication (in revised form) March 24, 1998; published electronically April 29, 1999.

http://www.siam.org/journals/sidma/12-2/31379.html

†Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213 (af1p@andrew.cmu.edu). The research of this author was partially supported by NSF grant 953074.

‡Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322, and Adam Mickiewicz University, Poznań, Poland (michal@mathcs.emory.edu, karonski@math.amu.edu.pl). The research of this author was partially supported by NSF grant INT-9406971 and KBN grant 2 P03A 023 09.

§DIMACS, Rutgers University, P.O. Box 1179, Piscataway, NJ 08855 (thoma@dimacs.rutgers.edu). This author was supported by DIMACS as a Postdoctoral Fellow. DIMACS is a cooperative project of Rutgers University, Princeton University, AT&T Research, Bellcore, and Bell Laboratories. DIMACS is an NSF Science and Technology Center, funded under contract STC-91-19999, and also receives support from the New Jersey Commission on Science and Technology. This author was also partially supported by NSF grant INT-9406971.

w.h.p. $\mathbf{SM}_n(4)$ has a Hamilton cycle, but the problem of whether or not $\mathbf{SM}_n(3)$ has w.h.p. a Hamilton cycle is one of the most important open questions in the theory of random graphs.

There is also a well-known bipartite mapping model $\mathbf{SM}_{n,n}(k)$. Walkup [12] had earlier shown that $\mathbf{SM}_{n,n}(2)$ has w.h.p. a perfect matching.

$\mathbf{ST}_n(k)$ is less well studied. Schmutz [9] computed the expected number of perfect matchings in $\mathbf{ST}_n(2)$ and showed that asymptotically it is $(4/e)^n$. He also studied a bipartite model $\mathbf{ST}_{n,n}(k)$ where the trees involved are random subtrees of the complete bipartite graph $K_{n,n}$ and showed that $\mathbf{ST}_{n,n}(2)$ has w.h.p. a perfect matching.

In section 2, we prove the following theorem.

THEOREM 1.2.

(a) $$\lim_{\substack{n\to\infty \\ n \text{ even}}} \operatorname{Prob}(\mathbf{ST}_n(1) \textit{ has a perfect matching}) = 0\,.$$

(b) $$\lim_{\substack{n\to\infty \\ n \text{ even}}} \operatorname{Prob}(\mathbf{ST}_n(2) \textit{ has a perfect matching}) = 1.$$

Using the proof methodology of Frieze and Łuczak [5], who showed that $\mathbf{SM}_n(5)$ has w.h.p. a Hamilton cycle, we prove a result on the existence of Hamilton cycles in $\mathbf{ST}_n(5)$ in section 3.

THEOREM 1.3.

$$\lim_{n\to\infty} \operatorname{Prob}(\mathbf{ST}_n(5) \textit{ has a Hamilton cycle}) = 1.$$

2. Perfect matchings—proof of Theorem 1.2.

(a) This follows immediately from Meir and Moon's result [8] that the size of the largest matching in a random tree is w.h.p. asymptotic to $(1-\rho)n \approx .432\, n$, where $\rho e^\rho = 1$.

(b) The proof of the second limit in Theorem 1.2 consists of several lemmas. Our starting point is a lemma by Gallai and Edmonds (Lemma 2.1) which gives a sufficient condition for the existence of a perfect matching. In the view of this lemma, it is enough to show that w.h.p. there is no bad set in $\mathbf{ST}_n(2)$. To show that we are going to distinguish different sizes of a bad set, Lemma 2.2 implies that for any fixed positive integer k_0 the sum of two random trees $\mathbf{ST}_n(2)$ has w.h.p. no bad set of size at most k_0. The next range of bad sets we eliminate are bad sets of size at most u_0n for some positive constant u_0. Using Lemma 2.4 we conclude that w.h.p. $\mathbf{ST}_n(2)$ has no such bad sets. Finally, a correspondence between labelled trees on n vertices and mappings from the set $[n]$ into itself and Lemma 2.7 imply that w.h.p. $\mathbf{ST}_n(2)$ does not contain a bad set of size larger than u_0n.

Before giving the lemmas we need some notation. Let $G=(V,E)$ be the graph. For $U\subseteq V$, let $G[U]=(U,E_U)$ be a subgraph of G induced on U, i.e., $E_U=\{e\in E;$ both vertices of e belong to $U\}$. Furthermore, let $N_G(U)=\{v\in V\setminus U;$ there is $u\in U$ such that $\{u,v\}\in E\}$ denote the neighborhood of the set U and set $N(U)=N_{\mathbf{ST}_n(2)}(U)$. A subset $U\subseteq V$ is said to be *stable* if $E_U=\emptyset$.

The following lemma is due to Gallai [6] and Edmonds [3] (cf. [4]).

LEMMA 2.1. *If a graph G does not have a perfect matching, then there exists $K\subseteq V(G), |K|=k\geq 0$ such that if $H=G[V(G)\setminus K]$, then*

(1) *H has at least $k+1$ components with an odd number of vertices*;

(2) *no odd component of H, which is not an isolated vertex, is a tree.*

The set K guaranteed by Lemma 2.1 will be called a *bad set.*

In the following sequence of lemmas, we are going to show that for n even $\mathbf{ST}_n(2)$ has w.h.p. no bad set.

Before starting with the lemmas, we recall the following two formulas: the number of forests on n vertices with k fixed roots is equal to kn^{n-k-1}, and the number of forests on n vertices with k roots (the roots can be any k of the n vertices) is equal to $\binom{n-1}{k-1}n^{n-k}$.

LEMMA 2.2. *For sets $K, L \subseteq V_n$, let $\mathcal{A}_1(K,L)$ be the event that $N(L) \subseteq K$. For positive integers k, l define the event $\mathcal{A}_1(k,l)$ by*

there exist $K, L \subseteq V_n, K \cap L = \emptyset, |K| = k, |L| = l$ such that $\mathcal{A}_1(K,L)$ occurs.

For $\epsilon, 0 < \epsilon < 1$, let $u(\epsilon) = [\frac{1-\epsilon}{3e^4(1+\epsilon)^{1+\epsilon}}]^{1/\epsilon}$ and suppose that $u = u(\epsilon)$ satisfies $(5e^4)^u/u^u \le 2^{1/8e^2}$.

Then setting $n_1 = \lfloor un \rfloor$ and $l_1 = \lceil (1+\epsilon)k \rceil$ and

$$\mathcal{A}_1(\epsilon) = \bigcup_{k=1}^{n_1} \bigcup_{l=l_1}^{\lfloor n/2 \rfloor} \mathcal{A}_1(k,l),$$

we have

$$\lim_{n\to\infty} \mathrm{Prob}(\mathcal{A}_1(\epsilon)) = 0.$$

Proof. To bound $\mathrm{Prob}(\mathcal{A}_1(k,l))$ we are going to divide the ranges of k and l into the following two cases:

(a) $l \le \lfloor n/(2e^2) \rfloor$ and any k,

(b) $l > \lfloor n/(2e^2) \rfloor$ and any k.

Fix K, L and the lowest numbered vertex $v \in K$. Now, each tree T with $N_T(L) \subseteq K$ is considered to be oriented towards v.

Case A. Let T be a tree oriented as described above. Delete edges oriented out of vertices in L. This leaves a forest F' with $l+1$ roots and n vertices. There are at most $(l+1)n^{n-l-2}$ such forests, (not all forests with $l+1$ roots and n vertices respect $N_T(L) \subseteq K$). To obtain T we construct a forest F'' with vertex set $K \cup L$ and roots K and take $T = F' \cup F''$. We can construct F'' in $k(k+l)^{l-1}$ ways. Hence,

$$\begin{aligned}
\mathrm{Prob}(\mathcal{A}_1(k,l)) &\le \binom{n}{k}\binom{n}{l}\left(\frac{(l+1)n^{n-l-2}k(k+l)^{l-1}}{n^{n-2}}\right)^2 \\
&\le \frac{(ne)^{k+l}}{k^k l^l} k^2 \frac{l^{2l}(1+\frac{k}{l})^{2l}}{n^{2l}} \\
&\le \frac{n^k k^2 e^{3k}}{k^k}\left(\frac{el}{n}\right)^l .
\end{aligned} \tag{3}$$

Putting $\mu_l = (el/n)^l$, we get $\mu_l/\mu_{l-1} < 1/2$ for $l < n/2e^2$.

Thus,

$$\sum_{l=l_1}^{\lfloor n/2e^2 \rfloor} \left(\frac{el}{n}\right)^l \le 2\left(\frac{el_1}{n}\right)^{l_1}.$$

Hence,

$$\sum_{k=1}^{n_1}\sum_{l=l_1}^{\lfloor n/2e^2\rfloor}\mathrm{Prob}(\mathcal{A}_1(k,l)) \le \sum_{k=1}^{\lfloor un\rfloor}\frac{n^k k^2 e^{3k}}{k^k}2\Big(\frac{el_1}{n}\Big)^{l_1}$$
$$\le 2\sum_{k=1}^{\lfloor un\rfloor}\Big(\frac{3e^4(1+\epsilon)^{1+\epsilon}k^{\epsilon}}{n^{\epsilon}}\Big)^k$$
$$= o(1).$$

Case B. Let T be a tree oriented as described above. Let F' be the forest obtained by deleting edges oriented out of K and deleting vertices in L. This forms a forest with $n-l$ vertices and k roots K. There are $k(n-l)^{n-l-k-1}$ such forests and each forest can be extended in at most $k(k+l)^{l-1}n^{k-1}$ ways to form the oriented tree T. Indeed, we attach the vertices from L by constructing a forest on $K\cup L$ with roots K in at most $k(k+l)^{l-1}$ ways. The remaining $k-1$ edges oriented out of K can be chosen in at most n^{k-1} ways. Hence,

$$\mathrm{Prob}(\mathcal{A}_1(k,l)) \le \binom{n}{k}\binom{n}{l}\Big[\frac{k(n-l)^{n-l-k-1}k(k+l)^{l-1}n^k}{n^{n-2}}\Big]^2$$
$$\le \frac{(ne)^{k+l}}{k^k l^l}\cdot\frac{k^4 e^{-2l+\frac{2l(l+k+1)}{n}}l^{2(l-1)}e^{2k}}{n^{2(l-1)}}$$
$$\le \frac{n^k e^{3k-l}k^4 e^{l+k+1}l^{l-2}}{k^k n^{l-2}}$$
$$= e\left(\frac{nk^{4/k}e^4}{k}\right)^k\left(\frac{l}{n}\right)^{l-2}.$$

For n large enough,

$$\sum_{l=\lfloor n/2e^2\rfloor+1}^{\lfloor n/2\rfloor}\Big(\frac{l}{n}\Big)^{l-2} \le n\Big(\frac{1}{2}\Big)^{\frac{n}{2e^2}-2} \le \Big(\frac{1}{2}\Big)^{\frac{n}{4e^2}}.$$

Hence,

$$\sum_{k=1}^{n_1}\sum_{l=\lfloor n/2e^2\rfloor+1}^{\lfloor n/2\rfloor}\mathrm{Prob}(\mathcal{A}_1(k,l)) \le e\Big(\frac{1}{2}\Big)^{\frac{n}{4e^2}}\sum_{k=1}^{\lfloor un\rfloor}\Big(\frac{nk^{4/k}e^4}{k}\Big)^k$$
$$\le e\Big(\frac{1}{2}\Big)^{\frac{n}{4e^2}}\sum_{k=1}^{\lfloor un\rfloor}\Big(\frac{5e^4 n}{k}\Big)^k$$
$$\le en\Big(\frac{(5e^4)^u}{2^{1/(4e^2)}u^u}\Big)^n$$
$$= o(1). \qquad \square$$

LEMMA 2.3. *Let ϵ be as in Lemma 2.2. Suppose a graph G contains a bad set $K, 1\le k=|K|\le u(\epsilon)n$, and no subset of K is bad. Let $H=G[V_n\setminus K]$ have $s\ge k+1$ odd components $C_1, C_2, \ldots, C_s$ with $n_1=n_2=\cdots=n_p=1<3\le n_{p+1}\le\cdots\le n_s$ vertices, respectively.*

Assume that $\mathcal{A}_1(\epsilon)$ does not occur. Then there exists a partition K, P, Q, R of V_n with $p = |P|, q = |Q|$ satisfying

$$N(R) \subseteq K, N(P) \subseteq K, N(Q) \subseteq K, \tag{4}$$

$$P \textit{ is a stable set}, \tag{5}$$

$$\textit{each vertex of } K \textit{ is adjacent to at least one member of } P \cup Q, \tag{6}$$

$$1 \le k \le u(\epsilon)n, 0 \le p + q < (1+\epsilon)k, p + \lfloor q/3 \rfloor \ge k \textit{ and } q = 0 \textit{ implies } p \ge k+1. \tag{7}$$

Proof. For the proof, see [4]. ∎

Let $\mathcal{A}_2(\epsilon)$ be the event that there is a partition satisfying (4)–(7) described in Lemma 2.3.

We can immediately show that for any fixed integer k_0

$$\lim_{\substack{n\to\infty \\ n\ even}} \text{Prob}(\mathbf{ST}_n(2) \text{ has a bad set } K, \text{ with } 1 \le |K| \le k_0) = 0.$$

Let us take $\epsilon = 1/2k_0$ and assume that $\mathcal{A}_1(\epsilon)$ does not occur. If there is a bad set K with $1 \le |K| \le k_0$, then the conditions of Lemma 2.3 are satisfied for some $k \le k_0$. But (7) implies $q < 3\epsilon k/2$ which in this case forces $q < 1$ or $q = 0$. But then $p \ge k+1$ contradicts $p < (1+\epsilon)k$.

In the proof of the following lemma we assume that $k \ge k_0$ for some suitably large k_0.

LEMMA 2.4. *For small ϵ*

$$\lim_{n\to\infty} \text{Prob}(\mathcal{A}_2(\epsilon)) = 0.$$

Proof. Fix K, P, Q, and $v \in K$. Each tree satisfying (4)–(6) can be chosen in at most $k(n-p-q)^{n-p-q-k-1} n^{k-1} k^p (k+q)^q$ ways. We first build a forest on $V \setminus (P \cup Q)$ with roots in K ($k(n-p-q)^{n-p-q-k-1}$ ways). Then each $x \in P$ is allowed to choose in K, each $y \in Q$ is allowed to choose in $K \cup Q$, and each $z \in K \setminus \{v\}$ is allowed to choose in V_n.

Let K_i be the set of vertices in K which have a neighbor in $P \cup Q$ in the tree T_i, $i = 1, 2$. There are two possibilities:

(a) $|K_1| \ge .9k$.

Of the $k^p(k+q)^q$ choices ascribed to vertices in $P \cup Q$, at most a proportion $.9^k$ will make $|K_1| \ge .9k$. Indeed, for each $x \in K$ the probability it is included in such a choice is at most

$$1 - \left(1 - \frac{1}{k}\right)^p \left(1 - \frac{1}{k+q}\right)^q \le .64,$$

for large enough k. The corresponding events for each x are clearly negatively correlated—note that we do not claim this for the choice of tree T_1, but only for the choices defined by the upper estimate. Thus,

$$\begin{aligned}\text{Prob}(|K_1| \ge .9k) &\le \binom{k}{.9k} (.64)^k \\ &\le (.9)^k.\end{aligned}$$

(b) $|K_1| < .9k$. By a similar argument,

$$\text{Prob}(K_2 \supseteq K \setminus K_1 \mid K_1,\, |K_1| < .9k) \le (.64)^{.1k}.$$

Combining the two cases we see that for $\delta = (.64)^{.1}$, we have

$$\begin{aligned}\text{Prob}(\mathcal{A}_2(k,p,q)) &\le 2\binom{n}{k,p,q}\Big[\frac{k(n-p-q)^{n-p-q-k-1}n^{k-1}k^p(k+q)^q}{n^{n-2}}\Big]^2\delta^k \\ &\le 2\frac{(ne)^{k+p+q}}{k^k p^p q^q}\cdot\frac{k^{2p+2q+2}e^{2q^2/k}e^{-\frac{p+q}{n}(n-p-q-k-1)}}{n^{2p+2q}}\delta^k,\end{aligned}$$

where $\mathcal{A}_2(k,p,q)$ is the event that there is a partition satisfying (4)–(7) in Lemma 2.3 for given k,p,q. We obtain for the probability of the event $\mathcal{A}_2(k,p,q)$, under the condition $q \ge 2$,

$$\begin{aligned}\text{Prob}(\mathcal{A}_2(k,p,q)) &\le 2\frac{e^{k-p-q}k^{2p+2q+2}}{k^k q^q p^p n^{p+q-k}}e^{2q^2/k}e^{\frac{2(1+\epsilon)(2+\epsilon)k^2+2(1+\epsilon)k}{n}}\delta^k \\ &\le 2\Big(\frac{k}{en}\Big)^{p+q-k}\Big(\frac{k}{p}\Big)^p\Big(\frac{k}{q}\Big)^q k^2 e^{2q^2/k}e^{\frac{2(1+\epsilon)(2+\epsilon)k^2+2(1+\epsilon)k}{n}}\delta^k.\end{aligned}$$

We continue with the bound on $\text{Prob}(\mathcal{A}_2(k,p,q))$, using $\frac{k}{p} \le 1+\frac{q}{p}$, $q \le \frac{3}{2}\epsilon k$, and $p+q-k \ge 1$. Furthermore, we use that the function x^{-x} on the interval $(0,\infty)$ has its maxima at $x = 1/e$. Thus, for every $k \ge k_0 = k_0(\epsilon)$

$$\begin{aligned}\text{Prob}(\mathcal{A}_2(k,p,q)) &\le 2\Big(\frac{k}{en}\Big)^{p+q-k}\Big(\frac{k}{p}\Big)^p\Big(\frac{k}{q}\Big)^q k^2 e^{9\epsilon^2 k/2}e^{12u(\epsilon)k}e^{4u(\epsilon)}\delta^k \\ &\le \frac{2k^3}{en}e^{4u(\epsilon)}\Big(\frac{2}{3\epsilon}\Big)^{3\epsilon k/2}e^{5\epsilon k+12u(\epsilon)k}\delta^k.\end{aligned}$$

Choose ϵ small enough such that $(2/3\epsilon)^{3\epsilon/2}e^{5\epsilon+12u(\epsilon)}\delta \le \mu < 1$ for some $0 < \mu < 1$. For $k \ge k_0$ let $\mathcal{S}(k) = \{(p,q);\ k,p,q \text{ satisfy conditions (7)}\}$. Note that $|\mathcal{S}(k)| \le 2k^2$ for ϵ small. We sum up over k, p, and q. Thus,

$$\begin{aligned}\text{Prob}(\mathcal{A}_2(\epsilon)) &= \sum_{k=k_0}^{\lfloor u(\epsilon)n\rfloor}\sum_{(p,q)\in\mathcal{S}(k)}\text{Prob}(\mathcal{A}_2(k,p,q)) \\ &\le \frac{4e^3}{n}\sum_{k=k_0}^{\lfloor u(\epsilon)n\rfloor}k^5\mu^k \\ &= o(1). \qquad \square\end{aligned}$$

Summing up, by choosing ϵ small enough and k_0 sufficiently large, so far we have proved that there is a constant $u_0 > 0$ such that

$$\lim_{\substack{n\to\infty \\ n\ even}} \text{Prob}(\mathbf{ST}_n(2) \text{ has a bad set } K, \text{ with } 1 \le |K| \le u_0 n) = 0.$$

To complete the proof of Theorem 1.2, we need to take care about large bad sets.

Lemma 2.5. *Let $\mathcal{A}_3$ denote the following event:*

$\mathbf{ST}_n(2)$ *contains at least* $(\log n)^3$ *sets* $S \subseteq V_n$ *satisfying*

$$|S| \leq \log\log n, \tag{8}$$

$$|E_S| \geq |S|. \tag{9}$$

Then $\lim_{n\to\infty} \mathrm{Prob}(\mathcal{A}_3) = 0$.

Proof. Fix $k \geq 2$. Let X_k be a random variable counting sets S with $|S| = k$ and $|E_S| \geq k$. Then

$$\begin{aligned} EX_k &\leq \binom{n}{k} \frac{\sum_{t=2}^{k} \sum_{\substack{i,j\geq 1\\ i+j=t}} (n-k)^t \left(\binom{k-1}{i-1} k^{k-i}\right)\left(\binom{k-1}{j-1} k^{k-j}\right)}{\left[(n-k)n^{n-(n-k)-1}\right]^2} \\ &= \binom{n}{k} \frac{k^{2k} \sum_{t=2}^{k} \binom{2(k-1)}{t-2} \left(\frac{n-k}{k}\right)^t}{[(n-k)n^{k-1}]^2} \\ &\leq e^k \left(\frac{n}{n-k}\right)^2 \left(\frac{k}{n}\right)^k \sum_{t=2}^{k} \binom{2(k-1)}{t-2} \left(\frac{n-k}{k}\right)^t. \end{aligned}$$

As we have $k \leq \log\log n$, we get

$$\begin{aligned} EX_k &\leq e^k \left(\frac{n}{n-k}\right)^2 k \binom{2k}{k} \left(\frac{n-k}{k}\right)^k \left(\frac{k}{n}\right)^k \\ &\leq (4e)^k k \left(\frac{n-k}{n}\right)^{k-2} \\ &\leq (4e)^k \log\log n. \end{aligned}$$

By the Markov inequality

$$\begin{aligned} \mathrm{Prob}(\mathcal{A}_3) &= \mathrm{Prob}\left(\sum_{k=3}^{\lfloor \log\log n \rfloor} X_k \geq (\log n)^3\right) \\ &\leq \frac{2\log\log n \cdot (4e)^{\log\log n}}{(\log n)^3} \\ &= o(1). \qquad \square \end{aligned}$$

Now we shall make use of the known one-to-one correspondence between the family of labelled trees on n vertices with two marked vertices and the family of functional digraphs $D(f)$ of mappings $f : [n] \to [n]$. Each such digraph D consists of vertices $S(f)$ which form cycles, and the remaining vertices form a set of trees $\mathcal{T}$ which are attached to the cycles. To obtain a tree T with two appropriately marked vertices from D, we shall consider vertices lying on the cycles as a permutation drawn in cyclic form. Next we write such a permutation in a line form, which in turn we treat as a directed path P. As a final step, we reattach the trees in $\mathcal{T}$ to their vertices on P to obtain a tree with two marked vertices (these two vertices are simply the beginning and the end of P). One can easily reproduce the correspondence from trees to mappings reversing the procedure described above. We believe that the one-to-one correspondence stated above is due to Joyal. A complete description of this correspondence can be found, for example, in Bender and Williamson [1].

This defines a natural measure preserving mapping ϕ from the space of random mappings to the space of random trees (ϕ just "forgets" the random choice of a pair of marked vertices). To finish the proof of Theorem 1.2 we will use ϕ to construct $\mathbf{ST}_n(2)$ in the following way: we first generate $\mathbf{SM}_n(2)$ from random functions f_1, f_2 and then apply ϕ to both of them.

DEFINITION 2.6. *Let a pair of sets $K, P \subseteq V_n$ be* matched *if*

(ι) P is stable in $\mathbf{ST}_n(2)$,

($\iota\iota$) $N(P) = K$,

($\iota\iota\iota$) $|P| \geq |K| - \delta(n)$,

where $\delta(n) = \lceil \frac{n}{\log\log n} + (\log n)^3 \rceil$.

LEMMA 2.7. *Suppose $\mathbf{ST}_n(2)$ has no bad sets of size $u_0 n$ or less but $\mathbf{ST}_n(2)$ contains a bad set $K_0, k = |K_0| > u_0 n$. Suppose K_0 does not strictly contain another bad set and $\mathcal{A}_3$ does not occur in $\mathbf{ST}_n(2)$. Let $S = S(f_1) \cup S(f_2)$. If $s = |S|$, then either $\mathbf{SM}_n(2)$ contains a matched pair K, P with*

$$|P| + \delta(n) + s \geq k \geq |K| \geq |P|$$

or

$$K_0 \text{ contains a bad set of } \mathbf{SM}_n(2).$$

Proof. Arguing as in Lemma 2.7 of [4] we see that $\mathbf{ST}_n(2)$ contains a matched pair K_1, P_1 with

$$|P_1| + \delta(n) \geq k \geq |K_1| \geq |P_1|.$$

Let $P = P_1 \setminus S$. Then P is stable and $|P| \geq |P_1| - s$. Also, $N_{\mathbf{SM}_n(2)}(P) \subseteq K_1$. Now take $K = N_{\mathbf{SM}_n(2)}(P)$. Either $|K| < |P|$ and K is a bad set of $\mathbf{SM}_n(2)$ or $|K| \geq |P|$ and K, P is the required matched pair. $\square$

Both possibilities in Lemma 2.7 are shown not to happen w.h.p. in [4], completing the proof of Theorem 1.2. (We observe first that w.h.p. $s = O(\sqrt{n})$ (cf. Kolchin [7]). The definition of a matched pair in [4] has to be amended to $\delta(n) + O(\sqrt{n})$, but this does not affect the proof there given in any significant way.)

3. Hamilton cycles—proof of Theorem 1.3. Frieze and Łuczak [5] proved that w.h.p. there is a Hamilton cycle in $\mathbf{SM}_n(5)$. We will use the same proof technique here, giving only a sketch as the main ideas are very similar.

We consider $\mathbf{ST}_n(5)$ to be the union of $\mathbf{ST}_n(4)$ and a random tree T_5. We observe first that Theorem 1.2 shows that w.h.p. $\mathbf{ST}_n(4)$ contains the union of two perfect matchings M_1, M_2. We can argue (see Lemma 2 of [5]) that M_1 and M_2 are an independent pair of matchings, chosen uniformly from the set of all possible perfect matchings. Furthermore, (see Lemma 3 of [5]) $M_1 \cup M_2$ is w.h.p. the union of at most $3 \log n$ vertex disjoint cycles—some cycles may possibly just be double edges.

We show next that w.h.p. $\mathbf{ST}_n(4)$ has good expansion properties. For sets $K, L \subseteq V_n$, let $\tilde{\mathcal{A}}_1(K, L)$ be the event that $N_{\mathbf{ST}_n(4)}(L) \subseteq K$ and let

$$\mathcal{A}_4 = \bigcup_{\substack{|K| \leq 10^{-3} n \\ |L| = 2|K|}} \tilde{\mathcal{A}}_1(K, L).$$

LEMMA 3.1. $\mathrm{Prob}(\mathcal{A}_4) = o(1)$.

Proof. It follows from (3) that

$$\begin{aligned}\mathrm{Prob}(\mathcal{A}_4) &\leq \sum_{k=1}^{10^{-3}n} \binom{n}{k}\binom{n}{2k}\left(\frac{(k+1)n^{n-k-2}2k(3k)^{k-1}}{n^{n-2}}\right)^4 \\ &\leq \sum_{k=1}^{10^{-3}n} (k+1)^4 \left(\frac{81e^3k}{4n}\right)^k \\ &= o(1). \qquad \square\end{aligned}$$

The idea now is to use the extension-rotation procedure (as described in [5]). The main idea that we get from [5] is to reserve the edges of T_5 for closing paths. More precisely, at some points of our extension-rotation procedure we will have a set A, $|A| \geq 10^{-3}n$ and for each $a \in A$ there is a collection of paths with endpoints $B(a)$, $|B(a)| \geq 10^{-3}n$, and we succeed if we always find a T_5-edge of the form (a, b) where $b \in B(a)$. With high probability we need only to attempt this at most $3\log n$ times (from Lemma 3.1). Let us suppose that the edges of T_5 come from a random mapping f_5, where an *adversary* has altered the edges coming out of a set S of $O(\sqrt{n})$ nodes. When given A, $\{B(a) : \ a \in A\}$ we choose the lowest numbered $a \in A \setminus S$ whose f_5 value has not been examined. So, w.h.p. we examine a further $O(\log n)$ a's before finding one with $f_5(a) \in B(a)$. Thus, w.h.p. the number of edges examined and altered throughout the procedure is $O(\sqrt{n})$ and we succeed in finding a Hamilton cycle.

REFERENCES

[1] E. A. Bender and S. G. Williamson, *Foundations of Applied Combinatorics*, Addison–Wesley, Reading, MA, 1991.

[2] C. Cooper and A. M. Frieze, *Hamilton cycles in random graphs and directed graphs*, to appear.

[3] J. Edmonds, *Paths, Trees, and Flowers*, Canad. J. Math., 17 (1965), pp. 449–467.

[4] A. M. Frieze, *Maximum matchings in a class of random graphs*, J. Combin. Theory Ser. B, 40 (1986), pp. 196–212.

[5] A. M. Frieze and T. Luczak, *Hamiltonian cycles in a class of random graphs: One step further*, in Random Graphs '87, M. Karonski, J. Jaworski, and A. Rucinski, eds., John Wiley, Chichester, 1990, pp. 53–59.

[6] T. Gallai, *Über extreme Punkt-und Kantenmengen*, Ann. Univ. Sci. Budapest Eötvös Sect. Math., 2 (1959), pp. 133–138.

[7] V. F. Kolchin, *Random Mappings*, Optimization Software Inc., New York, 1986.

[8] A. Meir and J. W. Moon, *The expected node-independence number of random trees*, Nederl. Akad. Wetensch. Proc. Ser. Indag. Math., 35 (1974), pp. 335–341.

[9] E. Schmutz, *private communication.*

[10] E. Schmutz, *Matchings in superpositions of (n, n)-bipartite trees*, Random Structures Algorithms, 5 (1994), pp. 235–241.

[11] E. Shamir and E. Upfal, *One-factor in random graphs based on vertex choice*, Discrete Math., 41 (1982), pp. 281–286.

[12] D. W. Walkup, *Matchings in random regular bipartite digraphs*, Discrete Math., 31 (1980), pp. 59–64.

SIAM J. DISCRETE MATH.
Vol. 12, No. 2, pp. 217–228

THE SIZE OF THE LARGEST COMPONENTS IN RANDOM PLANAR MAPS*

ZHICHENG GAO† AND NICHOLAS C. WORMALD‡

Abstract. Bender, Richmond, and Wormald showed that in almost all planar 3-connected triangulations (or dually, 3-connected cubic maps) with n edges, the largest 4-connected triangulation (or dually, the largest cyclically 4-edge-connected cubic component) has about $n/2$ edges [*Random Structures Algorithms,* 7 (1995), pp. 273–285]. In this paper, we derive some general results about the size of the largest component and apply them to a variety of types of planar maps.

Key words. planar map, 4-connected component, triangulation, cubic graph

AMS subject classifications. 05C30, 05C40

PII. S0895480195292053

1. Introduction.

Recently, Bender, Richmond, and Wormald [3] showed that in almost all 3-connected triangulations with n edges, the largest 4-connected triangulation has $n/2 + O(\lambda(n)n^{2/3})$ edges, for any function $\lambda(n) \to \infty$. (See section 3 for definitions.) Our objective in this paper is to abstract and generalize the technique of the above mentioned paper and apply it to a variety of types of planar maps. In principle, the general method applies to any combinatorial structure provided that the generating functions of the structures and their components have a relationship with each other which is common in enumerative map theory.

The next section contains analytic asymptotic results needed in a general form for evaluating the number of maps with a specified component of a given size, among other things. In section 3, we give some basic definitions for maps. The final section addresses the problem of transferring from a specified component to the largest component. The main result is Theorem 2, which determines almost surely the size of the largest component of a specified type in several classes of maps. Our results apply equally well to rooted or unrooted maps, though we work with the rooted ones in order to prove the main result.

2. General asymptotic results.

Throughout this paper, η and ρ are positive constants with $\rho < 1$. $\Omega(n)$ is any positive function which goes to ∞.

LEMMA 1. *Suppose $G_j \geq 0$ for $j \geq 0$ and $\sum_{j\geq 0} G_j\rho^j$ converges for some $\rho > 0$. Suppose there is a nonempty finite set J of indices satisfying $G_j > 0$ for $j \in J$ and $\gcd\{j - j' : j, j' \in J\} = 1$. Then, for any $0 < r \leq \rho$ and $0 < \theta < 2\pi$,*

$$\left|\sum_{j\geq 0} G_j \left(re^{i\theta}\right)^j\right| < \sum_{j\geq 0} G_j r^j.$$

*Received by the editors September 18, 1995; accepted for publication (in revised form) February 2, 1998; published electronically April 29, 1999.

http://www.siam.org/journals/sidma/12-2/29205.html

†Department of Mathematics and Statistics, Carleton University, Ottawa, Ontario K1S 5B6, Canada (zgao@math.carleton.ca). The research of this author was supported by NSERCC.

‡Department of Mathematics, University of Melbourne, Parkville, VIC 3052, Australia (nick@mundoe.maths.mu.oz.au). The research of this author was supported by the Australian Research Council.

Proof. Clearly, the left side does not exceed the right. Suppose there exist $0 < r \le \rho$ and $0 < \theta < 2\pi$ such that

$$\left| \sum_{j \ge 0} G_j (re^{i\theta})^j \right| = \sum_{j \ge 0} G_j r^j.$$

Then,

$$\sum_{j \ge 0} G_j \left(re^{i\theta}\right)^j = \sum_{j \ge 0} G_j r^j e^{i\theta'}$$

for some $0 \le \theta' < 2\pi$. This implies that

$$\sum_{j \ge 0} G_j r^j \left(1 - e^{i(j\theta - \theta')}\right) = 0,$$

and, hence,

$$j\theta - \theta' = 2k_j \pi$$

for some integers k_j and all $j \in J$. This gives

$$(j - j')\theta = 2(k_j - k_{j'})\pi \quad \text{for all} \quad j, j' \in J.$$

Since $\gcd\{j - j' : j, j' \in J\} = 1$, there are integers $m_{j,j'}$ such that

$$\sum_{j, j' \in J} m_{j,j'} (j - j') = 1.$$

Hence,

$$\theta = \sum_{j, j' \in J} m_{j,j'} (k_j - k_{j'}) 2\pi,$$

which contradicts the assumption that $0 < \theta < 2\pi$. □

THEOREM 1. *Suppose that* $G(z) = \sum_{n \ge 0} G_n z^n$ *satisfies the following conditions for some* $\rho > 0$ *and* $\eta > 0$*:*

(i) $G(z)$ *is analytic in* $\Delta = \{z : |z| \le \rho(1 + \eta)\} - [\rho, \rho(1 + \eta)]$.

(ii) *Let* $R = \sqrt{1 - z/\rho}$ *with the determination being positive for* $z < \rho$. *Then,*

$$G(z) = g_0 - g_2 R^2 + \sum_{j \ge 3} g_j R^j,$$

and the power series is convergent for $|z - \rho| \le \rho\eta$.

(iii) $G_n \ge 0$ *for all* $n \ge 0$, *and* $g_3 > 0$.

Then for $\beta = g_0/g_2$ *and any* $\Omega(n) \to \infty$ *as* $n \to \infty$ *with* $\Omega(n) = O(n^{1/3})$, *we have*

(a)

$$[z^n] G^j(z) \sim \frac{j g_3}{\Gamma(-3/2)} (n - j/\beta)^{-5/2} g_0^{j-1} \rho^{-n}$$

uniformly for all $1 \le j \le \beta n - n^{2/3} \Omega(n)$.

(b)

$$[z^n]G^j(z) = O\left(\rho^{-n} g_0^j n^{-2/3} \exp\left((n - j/\beta)n^{-2/3}\right)\right)$$

uniformly for all $j \geq \beta n + n^{2/3}\Omega(n)$, $j = O(n)$.

Proof. Let $\delta = n^{-2/3}/\Omega'(n)$, where $\Omega'(n) \to \infty$ and $\Omega'(n) = O(\Omega^{1/6}(n))$. Using conditions (i), (ii), and (iii) and Darboux's theorem (see Bender [1], for example), we have

$$G_n \sim \frac{g_3}{\Gamma(-3/2)} n^{-5/2} \rho^{-n}, \tag{1}$$

and, hence, $G_n > 0$ for all sufficiently large n. Therefore, again using (iii),

$$g_0 = G(\rho) > 0 \text{ and } g_2 = \rho G'(\rho) > 0, \tag{2}$$

and $G(z)$ satisfies the conditions of Lemma 1. Since $G(z)$ is continuous in Δ, using Lemma 1 we have that for any $\theta_0 > 0$, there exists $\eta' > 0$ such that

$$\left|G(\rho e^{i\theta})\right| \leq G(\rho)(1 - \eta') \tag{3}$$

for all $\theta_0 \leq \theta \leq 2\pi - \theta_0$. For sufficiently small θ_0 and ϵ, we define

$$\begin{aligned}
\Gamma_1 &= \{\rho + \epsilon e^{i\theta} : \ \pi/2 \leq \theta \leq 3\pi/2\},\\
\Gamma_2 &= \{\rho(1+x) + i\epsilon : \ 0 \leq x \leq \delta'\},\\
\Gamma_3 &= \{z : z = \rho(1+\delta)e^{i\theta}, \ 0 < \theta < \theta_0, \mathrm{Im}(z) \geq \epsilon\},\\
\Gamma_4 &= \{\rho(1+\delta)e^{i\theta} : \ \theta_0 \leq \theta \leq 2\pi - \theta_0\},\\
\Gamma_5 &= \{z : z = \rho(1+\delta)e^{i\theta}, \ 2\pi - \theta_0 < \theta < 2\pi, \mathrm{Im}(z) \leq -\epsilon\},\\
\Gamma_6 &= \{\rho(1+x) - i\epsilon : \ 0 \leq x \leq \delta'\},
\end{aligned}$$

where the direction of the contour is anticlockwise, and δ' is chosen so that $|\rho(1 + \delta') + i\epsilon| = \rho(1 + \delta)$. Let

$$I_l = \int_{\Gamma_l} G^j(z) z^{-(n+1)} dz$$

for $l = 1, \ldots, 6$. Then by Cauchy's formula,

$$[z^n]G^j(z) = \frac{1}{2\pi i}(I_1 + I_2 + I_3 + I_4 + I_5 + I_6). \tag{4}$$

We can assume $j = O(n)$. For $z = \rho(1+\delta)e^{i\theta} \in \Gamma_3$ and sufficiently small θ_0, we have

$$R = \sqrt{-\delta - (1+\delta)i\theta + O(\theta^2)}. \tag{5}$$

It follows from condition (ii) that

$$\begin{aligned}
|G(z)/g_0|^j &= |\exp(j \log(G(z)/g_0))|\\
&= \exp(j\delta/\beta + j(g_3/g_0)\mathrm{Re}(R^3) + jO(\theta^2 + \delta^2)).
\end{aligned} \tag{6}$$

For $n^{-2/3} \leq \theta \leq \theta_0$, and for sufficiently small θ_0 and sufficiently large n, we have from (5), and using $g_3 > 0$ from (2), that

$$\mathrm{Re}(R^3) = (-1/\sqrt{2})\theta^{3/2}(1 + O(\theta + \delta/\theta)) \leq (-1/2)\theta^{3/2},$$

and, hence,

$$|G(z)/g_0|^j = O(\exp(j\delta/\beta - (g_3/4g_0)j\theta^{3/2})).$$

For $0 \le \theta \le n^{-2/3}$, we have from (5) that $R^3 = O(1/n)$ and, hence, that

$$|G(z)/g_0|^j = O(\exp(j\delta/\beta)).$$

Therefore,

$$\begin{aligned} |I_3| &= O\left(\rho^{-n} g_0^j \exp((j/\beta - n)\delta)\right) \left(\int_0^{\theta_0} \exp(-(g_3/4g_0)j\theta^{3/2})d\theta + n^{-2/3}\right) \\ &= O\left(\rho^{-n} g_0^j j^{-2/3} \exp((j/\beta - n)\delta)\right). \end{aligned} \tag{7}$$

Assume $j \le \beta n - n^{2/3}\Omega(n)$. Using

$$\exp((j/\beta - n)\delta) = o\left((n - j/\beta)\delta\right)^{-3} = o\left((n - j/\beta)^{-5/2} n^{5/3}\right)$$

and

$$\exp((j/\beta - n)\delta) = o\left(n^{-10}\right) \quad \text{for } j \le \beta n/2,$$

we obtain

$$|I_3| = o\left(j\rho^{-n} g_0^j (n - j/\beta)^{-5/2}\right) \tag{8}$$

uniformly over j.

Using (3), we have, for sufficiently large n and $z \in \Gamma_4$, that

$$|G(z)| \le G(\rho) = g_0,$$

and, hence,

$$|I_4| = O(g_0^j \rho^{-n} (1+\delta)^{-n}) = O(g_0^j \rho^{-n} e^{-\delta n}). \tag{9}$$

It is clear that I_5 and I_6 are the negative conjugates of I_3 and I_2, respectively. All the error estimates above from (4)–(9) are independent of ϵ (or more precisely, uniform in the range $0 < \epsilon < \epsilon_0$, for some small but fixed positive constant ϵ_0). Hence, we can let $\epsilon \to 0$ in (4). Using (8) and (9), we note that I_1 goes to 0, and obtain

$$\begin{aligned} [z^n]G^j(z) &= (1/\pi)\mathrm{Im}\left(\lim_{\epsilon \to 0} I_2\right) \\ &\qquad + o\left(\rho^{-n} g_0^j j (n - j/\beta)^{-5/2}\right), \end{aligned} \tag{10}$$

provided the limit exists.

For $z = \rho(1+x) + i\epsilon \in \Gamma_2$, we have

$$\lim_{\epsilon \to 0} R = -ix^{1/2}, \tag{11}$$

and, hence, by condition (ii),

$$\begin{aligned} \lim_{\epsilon \to 0} G^j(z) &= g_0^j \exp(j(g_2/g_0)x + ij(g_3/g_0)x^{3/2} + O(jx^2)) \\ &= g_0^j \exp(jx/\beta)(1 + ij(g_3/g_0)x^{3/2} + O(jx^2)). \end{aligned}$$

Therefore, noting $\frac{dz}{dx} = \rho$ and $\lim_{\epsilon\to 0} \delta' = \delta$,

$$\begin{aligned} \operatorname{Im}\left(\lim_{\epsilon\to 0} I_2\right) &= \rho^{-n} g_0^j \int_0^\delta \exp(-(n-j/\beta)x)(jg_3/g_0)x^{3/2}(1+o(1))dx \\ &= \rho^{-n} g_0^{j-1} jg_3 \Gamma(5/2)(n-j/\beta)^{-5/2}(1+o(1)). \end{aligned} \tag{12}$$

Now part (a) follows from (10) and $\Gamma(5/2) = \pi/\Gamma(-3/2)$.

Now we prove part (b). Let

$$\Gamma_1 = \{z : z = \rho(1-n^{-2/3})e^{i\theta}, |\theta| \le \theta_0\},$$
$$\Gamma_2 = \{z : z = \rho(1-n^{-2/3})e^{i\theta}, \theta_0 \le |\theta| \le \pi\},$$

and define

$$I_l = \int_{\Gamma_l} G^j(z) z^{-(n+1)} dz$$

for $l = 1$ and 2. Then,

$$[z^n](G^j(z)) = \frac{1}{2\pi i}(I_1 + I_2). \tag{13}$$

Using Lemma 1, we have

$$|G(z)| \le g_0(1-\eta')$$

for $z \in \Gamma_2$ and some positive constant η'. Hence,

$$|I_2| = O\left(\rho^{-n} g_0^j (1-\eta')^j (1-n^{-2/3})^{-n}\right) = o\left(\rho^{-n} g_0^j (1-\eta'/2)^j\right) \tag{14}$$

uniformly for all $j \ge \beta n + n^{2/3}\Omega(n)$. Also, using an estimation similar to that for I_3 in part (a), this time with $\delta = -n^{-2/3}$, we again get the bound in (7); that is,

$$|I_1| = O\left(\rho^{-n} g_0^j n^{-2/3} \exp((n-j/\beta)n^{-2/3})\right) \tag{15}$$

uniformly for all $j \ge \beta n + n^{2/3}\Omega(n)$. Now part (b) follows from (13)–(15). □

We will say that $(A(z), B(z), C(z))$ is an *admissible triple* if it satisfies the following conditions:

H1: $A(z) = C(B(z))$;

H2: $A(z) = \sum_{n\ge 0} A_n z^n$ and $B(z) = \sum_{n\ge 0} B_n z^n$ are analytic in

$$\Delta_1 = \{z : |z| \le \rho(1+\eta), z \ne \rho, \operatorname{Arg}(z-\rho) \ne 0\},$$

and $C(z) = \sum_{n\ge 0} C_n z^n$ is analytic in

$$\Delta_2 = \{z : |z| \le B(\rho)(1+\eta), z \ne B(\rho), \operatorname{Arg}(z - B(\rho)) \ne 0\}$$

and $A_n,\ B_n,\ C_n \ge 0$ for all n;

H3:

$$A(z) = a_0 - a_2 R^2 + \sum_{j\ge 3} a_j R^j, \quad B(z) = b_0 - b_2 R^2 + \sum_{j\ge 3} b_j R^j,$$

where $R = \sqrt{1 - z/\rho}$ (the determination being positive for $z < \rho$) and the power series in R has positive radii of convergence, and

$$C(z) = c_0 - c_2 S^2 + \sum_{j \geq 3} c_j S^j,$$

where $S = \sqrt{1 - z/b_0}$ (the determination being positive for $z < b_0$) and the power series in S has a positive radius of convergence;
H4: $a_3 > 0$, $b_3 > 0$, and $c_3 > 0$.

LEMMA 2. *Let $(A(z), B(z), C(z))$ be an admissible triple, $\beta = b_0/b_2$, and $\Omega(n)$ be any function which goes to positive infinity. Then,*

$$\alpha_n = \sum_{|j-\beta n| < n^{2/3}\Omega(n)} C_j [z^n] B^j(z)/A_n = 1 - \frac{a_2 b_3}{a_3 b_2} + o(1). \tag{16}$$

Proof. Using Darboux's theorem [1], or Theorem 1(a) with $j = 1$, we have

$$\begin{aligned} A_n &\sim (a_3/\Gamma(-3/2))\, n^{-5/2} \rho^{-n}, \\ B_n &\sim (b_3/\Gamma(-3/2))\, n^{-5/2} \rho^{-n}, \\ C_n &\sim (a_3/\Gamma(-3/2))\, n^{-5/2} b_0^{-n}. \end{aligned}$$

By Theorem 1(a), we have

$$[z^n] B^j(z) = \frac{j b_3 b_0^{j-1}}{\Gamma(-3/2)} (n - j/\beta)^{-5/2} \rho^{-n} \tag{17}$$

uniformly for all $j \leq \beta n - n^{3/2}\Omega(n)$. Therefore,

$$\begin{aligned} &\sum_{j \leq \beta n - n^{2/3}\Omega(n)} C_j\, [z^n] B^j(z)/A_n \qquad (18) \\ &= (b_3/a_3) \sum_{j \leq n/\log n} C_j j b_0^{j-1} (1 + o(1)) + \sum_{n/\log n < j \leq \beta n/2} O(j^{-3/2}) \\ &\quad + \sum_{\beta n/2 < j \leq \beta n - n^{2/3}\Omega(n)} O\left(j^{-3/2} (n - j/\beta)^{-5/2}\; n^{5/2}\right) \\ &= (b_3/a_3) C'(b_0) + O\left(n \int_{\beta n/2}^{\beta n - n^{2/3}\Omega(n)} (n - x/\beta)^{-5/2} dx\right) + o(1) \\ &= (b_3/a_3) C'(b_0) + O(\Omega^{-3/2}(n)) + o(1). \end{aligned}$$

Since $C'(b_0) = a_2/b_2$, we have

$$\sum_{j \leq \beta n - n^{2/3}\Omega(n)} C_j [z^n] B^j(z)/A_n = a_2 b_3/(a_3 b_2) + o(1). \tag{19}$$

By Theorem 1(b), we have

$$\begin{aligned} &\sum_{j \geq \beta n + n^{2/3}\Omega(n)} C_j\, [z^n] B^j(z)/A_n \\ &= \sum_{j \geq \beta n + n^{2/3}\Omega(n)} O\left(n^{-2/3} \exp\left((n - j/\beta) n^{-2/3}\right)\right) \\ &= O\left(\exp\left(-\Omega(n)/\beta\right)\right) \\ &= o(1). \qquad (20) \end{aligned}$$

Now the lemma follows from H1, (19), and (20). ∎

LEMMA 3. *Let $(A(z), B(z), C(z))$ be an admissible triple, α_n and $\beta = b_0/b_2$ as defined in Lemma 2. If $B(z) = zh(A(z))$ for some function $h(z)$, which is analytic in $\{z : |z| \leq a_0\}$, then*

$$\alpha_n = \beta + o(1).$$

Proof. Let

$$h(z) = \sum_{k \geq 0} h_k (z - a_0)^k.$$

Then,

$$\begin{aligned} B(z) &= z \sum_{k \geq 0} h_k \left(-a_2 R^2 + \sum_{j \geq 3} a_j R^j \right)^k \\ &= \rho(1 - R^2)(h_0 - h_1 a_2 R^2 + h_1 a_3 R^3 + \text{higher power terms in } R). \end{aligned}$$

Therefore,

$$b_0 = \rho h_0, \quad b_2 = \rho(h_0 + h_1 a_2), \quad b_3 = \rho h_1 a_3. \tag{21}$$

Thus, we have

$$1 - \frac{a_2 b_3}{a_3 b_2} = h_0/(h_0 + h_1 a_2) = \beta.$$

Now the lemma follows from Lemma 2. ∎

3. General map definitions. A *planar map* is a connected graph G embedded in the sphere S such that all components of $S - G$ are simply connected regions, which are called *faces*. Loops and multiple edges are permitted in G. In this paper a planar map is simply a *map*. A map is *rooted* if an edge is distinguished, together with a vertex incident with the edge and a side of the edge. This is useful in enumeration, since we regard two maps to be equivalent if there is a homeomorphism of the sphere which takes one to the other; for rooted maps, the homeomorphism must preserve the rooting.

A graph (or the corresponding map) is *k-connected* if it has at least k vertices, requires removing at least k vertices to separate the graph, and the graph has no loops if $k \geq 2$ and no multiple edges if $k \geq 3$.

A *triangulation* is a map in which all the faces are triangles. In this paper, a *quadrangulation* is a map which has no multiple edges and in which all the faces are quadrangles. *Cubic* maps have all vertices of degree 3 (the duals of triangulations), and *bicubic* maps are cubic maps whose graphs are bipartite.

To *contract* a loop e of a map with respect to a face F, shrink e and the part of the sphere on the other side of e from F down to the vertex incident with e. A *loopless component* in a map is a map obtained by specifying a face F and then contracting all loops with respect to F. A *simple component* in a loopless map is defined similarly by contracting all 2-cycles into single edges, with respect to some face. The same definition serves for a 3-connected component of a 2-connected triangulation, since in such a map, a 2-vertex cut determines a 2-cycle. A quadrangulation is called *simple* if all 4-cycles are facial. A 2-connected component of a map is simply the map induced by a 2-connected component (maximal 2-connected subgraph) of its graph. To treat 3-connected components in 2-connected maps, we use the one-to-one correspondence between rooted quadrangulations and rooted maps [6].

4. Components in planar maps. In this section, we apply Theorem 1 to obtain the size of the largest component of the following types in the specified families of planar maps. In some cases the components are not actually submaps but are the components in a unique decomposition of the maps under some natural definition of decomposition. It could also be applied to obtain the analogous result about 4-connected components in 3-connected triangulations obtained in [3].

1. 2-connected component in a general map;
2. loopless component in a general map, or dually, 2-edge connected component in a general map;
3. simple component in a loopless map, or dually, 3-edge connected component in a 2-edge connected map;
4. 3-connected component in a 2-connected triangulation;
5. 3-connected bicubic component in a 2-connected bicubic map;
6. 3-connected component in a 2-connected map, or, equivalently, simple component in a quadrangulation;

The following lemma is useful; the proof of a special case can be found in [7].

LEMMA 4. *In a family of rooted maps of any of the types listed above, for any* $\epsilon > 0$*, almost all maps with* n *edges have at most one component that has more than* $n^{2/3+\epsilon}$ *edges.*

Proof. For any of these families of maps (bicubic maps [7], other maps [5]), the number of maps on n edges is asymptotic to

$$M(c,r,n) = cn^{-5/2}r^n$$

for some constants c and r, depending on the family, where for some of the families n is restricted in a natural way to a subset of the positive integers. (For example, in bicubic maps, n is a multiple of 3.)

We now treat the family of all maps in detail; the other cases are almost identical. Suppose a rooted map R with n edges has at least two 2-connected components containing more than $n^{2/3+\epsilon}$ edges. Then, there is some cut-vertex at which the map splits into two submaps, R_1 and R_2, containing at least $n^{2/3+\epsilon}$ edges each. One of these (say R_1) contains the root edge of R. Given R_1 with, say, $n-k$ edges, there are at most $2(n-k)$ places at which R_2 could be attached. Thus, summing over k, the number of possibilities for R is at most

$$\sum_{k=\lceil n^{2/3+\epsilon}\rceil}^{\lfloor n-n^{2/3+\epsilon}\rfloor} O(1)(n-k)k^{-5/2}(n-k)^{-5/2}r^n = O(1)r^n n^{-3/2}\int_{n^{2/3+\epsilon}}^{n/2} k^{-5/2}dk$$

$$= O(n^{-3\epsilon/2})M(c,r,n). \qquad \square$$

Suppose now that we have specified a family $\mathcal{A}$ of maps and a family $\mathcal{C}$ of components. A map in $\mathcal{A}$ is called $\mathcal{C}$*-rooted* (*component rooted*) if a component in $\mathcal{C}$ is distinguished together with a rooting of the component as a map. Let $A(z)$ be the generating function of the $\mathcal{C}$-rooted maps in $\mathcal{A}$, and let $C(z)$ be the generating function for the rooted maps in $\mathcal{C}$. Then, in general, there will be a function $B(z)$ such that H1 holds. (In general, a member of the maps in $\mathcal{A}$ can be created by taking one of the components in $\mathcal{C}$ and replacing some parts of the component, perhaps each face, or each nonroot edge, for instance, by something very close to one of the maps in question. $C_k B^k(z)$ counts the number of $\mathcal{C}$-rooted maps whose root components have size k.) Then, $(A(z), B(z), C(z))$ will be an admissible triple (provided the other requirements H2, H3, and H4 hold), and so Lemmas 2 and 3 will apply.

The following lemma will be used to pass from the root component size to the largest component size. Here, "size" refers to the number of edges.

LEMMA 5. *Let $M(z)$ be the generating function for rooted maps of one of the above-listed families $\mathcal{A}$ by edges, $C(z)$ the generating function for the corresponding family $\mathcal{C}$ of components, and $A(z)$ the generating function for the $\mathcal{C}$-rooted maps in $\mathcal{A}$. Assume that $(A(z), B(z), C(z))$ is an admissible triple for some function $B(z)$. Let X_n be the size of a largest component in a random map with uniform distribution in the family. Let α_n and β be defined as in Lemma 2. Then,*

$$\mathbf{P}\left(|X_n - \beta n| < n^{2/3}\Omega(n)\right) = \alpha_n \frac{A_n}{\beta M_n} + o(1).$$

Proof. A map in $\mathcal{A}$ is called *doubly rooted* if it is $\mathcal{C}$-rooted and has an additional secondary rooting (that is, an edge, and a side and end of that edge are distinguished, perhaps the same as the primary rooting). Let $\mathcal{D}_n$ be the set of doubly rooted maps in $\mathcal{A}$ that have n edges, in which the root component has size k with $|k - \beta n| < n^{2/3}\Omega(n)$.

By the definition of α_n,

$$|\mathcal{D}_n| = 4n\alpha_n A_n. \tag{22}$$

On the other hand, let $\mathcal{Z}_n$ be the subset of $\mathcal{D}_n$ in which all rooted maps have at least two components of size greater than $\beta n - n^{2/3}\Omega(n)$. Since each rooted map in $\mathcal{Z}_n$ can be $\mathcal{C}$-rooted in at most $O(n)$ ways, it follows from Lemma 4 that

$$|\mathcal{Z}_n| = o(1)nM_n. \tag{23}$$

Hence,

$$|\mathcal{D}_n| = (4\beta n + o(n))\mathbf{P}(|X_n - \beta n| < n^{2/3}\Omega(n))M_n + o(1)nM_n. \tag{24}$$

Now the lemma follows from (22)–(24). ∎

Now we prove the following main result.

THEOREM 2.

(1) *Let X_n be the number of edges of a largest 2-connected component in a random rooted map with n edges. Then,*

$$\mathbf{P}(|X_n - n/3| < n^{2/3}\Omega(n)) = 1 + o(1).$$

(2) *Let X_n be the number of edges of a largest simple component in a random rooted loopless map with n edges. Then,*

$$\mathbf{P}(|X_n - 2n/3| < n^{2/3}\Omega(n)) = 1 + o(1).$$

(3) *Let X_n be the number of edges of a largest 3-connected component in a random rooted 2-connected triangulation with $3n$ edges. Then,*

$$\mathbf{P}(|X_n - 3n/2| < n^{2/3}\Omega(n)) = 1 + o(1).$$

(4) *Let X_n be the number of edges of a largest* 3*-connected component in a random rooted* 2*-connected bicubic map with* $3n$ *edges. Then,*

$$\mathbf{P}(|X_n - 5n/17| < n^{2/3}\Omega(n)) = 1 + o(1).$$

(5) *Let X_n be the number of edges of a largest loopless component in a random rooted map with n edges. Then,*

$$\mathbf{P}(|X_n - 2n/3| < n^{2/3}\Omega(n)) = 1 + o(1).$$

(6) *Let X_n be the number of edges of a largest* 3*-connected component in a random rooted* 2*-connected map with n edges. Then,*

$$\mathbf{P}(|X_n - n/3| < n^{2/3}\Omega(n)) = 1 + o(1).$$

Proof. We note for the map families and component families in (1)–(4), the $\mathcal{C}$-rooted maps are the same as rooted maps, since there is exactly one component containing the root. Therefore, in the terminology of Lemma 5, $M_n = A_n$ for these maps. Also, as shown below, the conditions of Lemma 3 are also satisfied by these maps. It will then follow that (1)–(4) hold by Lemmas 2, 3, and 5.

For part (1), let A_n (C_n) be the number of rooted maps (2-connected maps) with n edges. Define

$$A(z) = \sum_{n\geq 1} A_n z^n, \quad C(z) = \sum_{n\geq 1} C_n z^n, \quad B(z) = z(1 + A(z))^2.$$

It follows from [4] that $(A(z), B(z), C(z))$ is an admissible triple. Using (4.2) of [4] with a bit of calculation, we have

$$b_0 = 4/27, \quad b_2 = 12/27.$$

Hence, $\beta = 1/3$.

For part (2), let A_n (C_n) be the number of loopless rooted maps (simple maps) with n edges. Define

$$A(z) = \sum_{n\geq 1} A_n z^n, \quad C(z) = \sum_{n\geq 1} C_n z^n, \quad B(z) = z(1 + A(z)).$$

Noting that simple maps are obtained by closing digons in loopless maps, it is easily seen that $(A(z), B(z), C(z))$ is an admissible triple. (See [9] for more details.) Using (20) and (22) of [5] with a bit of calculation, we have

$$b_0 = 1/8, \quad b_2 = 3/16.$$

Hence, $\beta = 2/3$.

For part (3), let A_n (C_n) be the number of rooted 2-connected triangulations (3-connected triangulations) with $3n$ edges. Define

$$A(z) = \sum_{n\geq 1} A_n z^n, \quad C(z) = \sum_{n\geq 1} C_n z^n, \quad B(z) = z(1 + A(z))^3.$$

Then $(A(z), B(z), C(z))$ is an admissible triple [9]. Using (15) and (16) of [9], we have

$$z = \theta(1 - 2\theta)^2, \quad B(z) = \theta(1 - 3\theta)^3(1 - 2\theta)^{-4}.$$

After a bit of calculation, we obtain

$$B(z) = 27/256 - (27/128)(1 - 27z/2) + (3\sqrt{3}/32)(1 - 27z/2)^{3/2} + \cdots$$

and, hence, $\beta = b_0/b_2 = 1/2$.

For part (4), let A_n (C_n) be the number of rooted 2-connected bicubic maps (3-connected bicubic maps) with $3n$ edges. Define

$$A(z) = \sum_{n\geq 1} A_n z^n, \quad C(z) = \sum_{n\geq 1} C_n z^n, \quad B(z) = z(1 + A(z))^3.$$

Then $(A(z), B(z), C(z))$ is an admissible triple [8]. Using (21) of [8] with a bit of calculation, we obtain

$$B(z) = 125/512 - (425/512)(1 - 8z) + (75/64)(1 - 8z)^{3/2} + \cdots$$

and, hence, $\beta = 5/17$.

The proofs of (5) and (6) are a bit different because $M(z) \neq A(z)$ in these cases.

For part (5), let M_n (C_n) be the number of all rooted maps (loopless maps) with n edges, and let A_n be the number of component rooted maps with n edges. Define

$$M(z) = \sum_{n\geq 0} M_n z^n, \quad A(z) = \sum_{n\geq 1} A_n z^n, \quad C(z) = \sum_{n\geq 1} C_n z^n.$$

If we let $L(z)$ be the generating function for those maps whose root faces consist of only loops, including the single vertex map, then it is easy to show that

$$L(z) = \sum_{j\geq 0} (zM(z))^j = 1/(1 - zM(z)), \tag{25}$$

$$A(z) = C(zL^2(z)), \tag{26}$$

$$A(z) = M(z) - 1 - zM^2(z). \tag{27}$$

Let $B(z) = zL^2(z)$; then $(A(z), B(z), C(z))$ is an admissible triple. Using (4.20) of [2] with a bit of algebra, we obtain

$$M(z) = 4/3 - (4/3)(1 - 12z) + (8/3)(1 - 12z)^{3/2} + \cdots \tag{28}$$

$$A(z) = 32/27 - (8/9)(1 - 12z) + (56/27)(1 - 12z)^{3/2} + \cdots \tag{29}$$

$$B(z) = 27/256 - (81/512)(1 - 12z) + (27/512)(1 - 12z)^{3/2} + \cdots \tag{30}$$

Hence, $\beta = b_0/b_2 = 2/3$ and $\alpha_n = 6/7 + o(1)$. It follows from Lemma 5 that

$$\mathbf{P}(|X_n - 2n/3| < n^{2/3}\Omega(n)) = (9/7)(56/27)(3/8) + o(1) = 1 + o(1).$$

For part (6), let M_n (C_n) be the number of rooted quadrangulations (simple quadrangulations) with n faces (i.e., $2n$ edges). Define

$$M(z) = \sum_{n\geq 2} Q_n z^n, \quad C(z) = \sum_{n\geq 4} C_n z^n, \quad B(z) = M(z)/z.$$

Let $A(z)$ be the generating function for component rooted quadrangulations such that the root component has at least four faces. Using the argument of [6], we have

$$A(z) = \sum_{j\geq 4} C_j B^j(z) = C(B(z)). \tag{31}$$

Note that for $F(x, y)$ and $Q_N^*(x, y)$ defined in [6], we have

$$M(z) = F(z, z), \quad C(z) = Q_N^*(z, z).$$

It follows from [6] that

$$\sum_{j \geq 4} C_j B^{j-1}(z) = B(z)(1 - B(z))/(1 + B(z)) - z, \tag{32}$$

and $M(z)$ is given parametrically by

$$M(z) = t^2(1 - t), \quad z = t(1 - t)^2. \tag{33}$$

Letting $R = (1 - 27z/4)^{1/2}$, we obtain the following expansions:

$$M(z) = 1/27 - (4/27)R^2 + (8/27\sqrt{3})R^3 + \cdots \tag{34}$$

$$B(z) = 1/4 - (3/4)R^2 + (2/\sqrt{3})R^3 + \cdots. \tag{35}$$

Using (31), (32), and (35), we obtain

$$A(z) = 1/2160 - (91/5400)R^2 + (97/675\sqrt{3})R^3 + \cdots. \tag{36}$$

Therefore,

$$\beta = 1/3, \quad \alpha_n = 200/291 + o(1),$$

and, hence,

$$\frac{\alpha_n A_n}{\beta M_n} = 1 + o(1).$$

Now part (6) follows from Lemma 5 and the correspondence between quadrangulations and maps. □

Since almost all rooted maps in the families we consider have no symmetries [8], we immediately get the following.

COROLLARY. *The conclusions of Theorem* 2 *also apply for random unrooted maps.*

REFERENCES

[1] E. A. BENDER, *Asymptotic methods in enumeration*, SIAM Rev., 16 (1974), pp. 485–515.

[2] E. A. BENDER, E. R. CANFIELD, AND R. W. ROBINSON, *The enumeration of maps on the torus and the projective plane*, Canad. Math. Bull., 31 (1988), pp. 257–271.

[3] E. A. BENDER, L. B. RICHMOND, AND N. C. WORMALD, *Largest* 4*-connected components of* 3*-connected planar triangulations*, Random Structures Algorithms, 7 (1995), pp. 273–285.

[4] E. A. BENDER AND N. C. WORMALD, *The asymptotic number of rooted nonseparable maps on a surface*, J. Combin. Theory Ser. A, 49 (1988), pp. 370–380.

[5] Z. C. GAO, *A pattern for the asymptotic number of rooted maps on surfaces*, J. Combin. Theory Ser. A, 64 (1993), pp. 246–264.

[6] R. C. MULLIN AND P. J. SCHELLENBERG, *The enumeration of c-nets via quadrangulations*, J. Combin. Theory, 4 (1968), pp. 259–276.

[7] L. B. RICHMOND AND N. C. WORMALD, *Random triangulations of the plane*, European J. Combin., 9 (1988), pp. 61–71.

[8] L. B. RICHMOND AND N. C. WORMALD, *Almost all maps are asymmetric*, J. Combin. Theory Ser. B, 63 (1995), pp. 1–7.

[9] W. T. TUTTE, *The enumerative theory of planar maps*, in A Survey of Combinatorial Theory, J.N. Srivastava et al., eds., North–Holland, Amsterdam, 1973, pp. 437–448.

SIAM J. Discrete Math.
Vol. 12, No. 2, pp. 229–242

GRAPH EAR DECOMPOSITIONS AND GRAPH EMBEDDINGS*

JIANER CHEN† AND SAROJA P. KANCHI‡

Abstract. Ear decomposition of a graph has been extensively studied in relation to graph connectivity. In this paper, a connection of ear decomposition to graph embeddings is exhibited. It is shown that constructing a maximum-paired ear decomposition of a graph and constructing a maximum-genus embedding of the graph are polynomial-time equivalent. Applications of this connection are discussed.

Key words. graph connectivity, graph embedding, ear decomposition, graph maximum genus, algorithm

AMS subject classifications. 05C10, 05C40, 05C85, 57M15, 68R10

PII. S0895480196304234

1. Introduction.

An ear decomposition of a graph is a way of partitioning the edge set of the graph into an ordered collection of edge-disjoint simple paths called ears. It is well known that a graph has an ear decomposition if and only if it is 2-edge-connected [21].

Ear decomposition of a graph has received considerable attention recently because of its close relation to graph connectivity. Lovasz [15] first noted that ear decompositions can be found quickly in parallel. Ear decompositions have been used in designing efficient sequential and parallel algorithms for 2-edge-connectivity, 2-vertex-connectivity, 3-vertex-connectivity [17], and 4-vertex-connectivity [14].

Variations of ear decomposition of a graph have also been proposed. The concept of 2-connected semisimplicial ear decomposition was introduced in the study of average genus of 2-connected simple graphs [4]. Algorithms for efficiently constructing 3-connected ear decompositions have been discussed [8]. Cheriyan and Maheshwari [9] used nonseparating ear decomposition to develop efficient algorithms for finding independent spanning trees in a graph. Ear decompositions for 4-connected graphs have also been studied [7, 20].

The *maximum genus* $\gamma_M(G)$ of a graph G is defined to be the maximum integer k such that there exists a cellular embedding of G into the orientable surface of genus k. Since the introductory investigation by Nordhaus, Stewart, and White [16], maximum genus embeddings of a graph have been extensively studied. (For a survey, see Ringeisen [18].) Certain graph classes can be precisely characterized by their maximum genus. For example, a graph has maximum genus 0 if and only if it is a cactus [18], and a 2-edge-connected graph has maximum genus 1 if and only if it is a necklace (with five exceptions) [5]. Recent investigations on maximum genus have focused on developing efficient algorithms for maximum genus embeddings of a graph. A polynomial-time algorithm for constructing a maximum genus embedding

*Received by the editors May 24, 1996; accepted for publication (in revised form) December 23, 1998; published electronically April 29, 1999. This research was supported in part by National Science Foundation grants CCR-9110824 and CCR-9613805.

http://www.siam.org/journals/sidma/12-2/30423.html

†Department of Computer Science, Texas A&M University, College Station, TX 77843-3112 (chen@cs.tamu.edu).

‡Department of Science and Mathematics, Kettering University, Flint, MI 48504 (skanchi@kettering.edu). The research of this author was partially supported by the Engineering Excellence Award from Texas A&M University.

of a graph was developed by Furst, Gross, and McGeoch [10] based on a characterization of the maximum genus of a graph given by Xuong [22] and an efficient matroid parity algorithm by Gabow and Stallmann [11]. A linear-time algorithm for constructing a maximum genus embedding of a graph of bounded maximum genus has been developed by Chen [1].

In this paper, we exhibit an interesting connection between ear decompositions and maximum genus embeddings of a graph. We introduce the concept of maximum-paired ear decomposition of a graph (the precise definition will be given in section 2). We prove that a maximum-paired ear decomposition of a graph G is k-paired if and only if the maximum genus of the graph G is k. Then we show by developing efficient algorithms that constructing a maximum-paired ear decomposition of a graph is polynomial-time equivalent to constructing a maximum genus embedding of the graph. Since a maximum genus embedding of a graph can be constructed in polynomial time [10], our results imply a polynomial-time algorithm for constructing a maximum-paired ear decomposition of a graph. Applications of this connection will also be discussed.

We point out that our results lose no generality with the restriction that the graph has an ear decomposition, i.e., the graph should be 2-edge-connected. In fact, the maximum genus of a graph is equal to the sum of maximum genera of its 2-edge-connected components [5]. Thus, our results can be applied directly to each 2-edge-connected component of a graph if it is not 2-edge-connected.

The paper is organized as follows. Section 2 reviews the related algorithmic and topological preliminaries and definitions. In section 3, the relationship between ear decompositions and Xuong trees, which are a kind of spanning tree closely related to maximum genus embeddings, is investigated. Section 4 studies the algorithmic relationship between maximum genus embeddings and Xuong trees. Combining the results of sections 3 and 4, section 5 concludes with the algorithmic relationship between maximum genus embeddings and maximum-paired ear decompositions. Some applications of this relationship are also described.

2. Preliminaries and definitions. It is assumed that the reader is somewhat familiar with the fundamentals of graph embeddings. For further description, see Gross and Tucker [12].

A *graph* may have multiple adjacencies or self-adjacencies. The topological spaces discussed in this paper are all subspaces of the 3-dimensional space. An *orientable surface* is a closed and connected 2-dimensional manifold that does not contain a Möbius band. It is well known that every orientable surface is homeomorphic to a generalized torus S_g for some integer $g \geq 0$, where S_g is obtained from a sphere by adding g handles. The integer g is called the *genus* of the surface S_g. A sphere, for example, is a surface of genus 0, a torus is a surface of genus 1, and a two-handled torus is a surface of genus 2. An *embedding* of a graph G to a surface S is a continuous one-to-one mapping. The embedding is *cellular* if each component of $S-G$ is homeomorphic to an open disk. All our embeddings in this paper are cellular.

A *rotation* at a vertex v is a cyclic permutation of the edge-ends incident on v. A list of rotations, one for each vertex of the graph, is called a *rotation system* of the graph.

An embedding of a graph G in an orientable surface induces a rotation system as follows: The rotation at vertex v is the cyclic permutation corresponding to the order in which the edge ends are traversed in an orientation-preserving tour around v. Conversely, by the Heffter–Edmonds principle, every rotation system induces a unique embedding of G into an orientable surface (see [12]). This bijectivity enables us to study graph embeddings based on graph rotation systems. We will interchangeably use the phrases "an embedding of a graph" and "a rotation system of a graph." In particular, if $\rho(G)$ is a rotation system of a graph G, we will denote by $\gamma(\rho(G))$ the genus of the corresponding embedding of the graph G. Moreover, there is a linear-time algorithm that, given a rotation system $\rho(G)$ of a graph $G = (V, E)$, traces the boundary walks of all faces in the rotation system $\rho(G)$ [6]. Thus, the genus $\gamma(\rho(G))$ of the rotation system $\rho(G)$ can be calculated in linear time using the *Euler polyhedral equation* [12]:

$$|V| - |E| + |F| = 2 - 2\gamma(\rho(G)),$$

where F is the set of faces in the rotation system $\rho(G)$.

Let T be a spanning tree of a graph G. The edge complement $G - T$ will be called a *cotree*. The number of edges in any cotree is known as the *cycle rank* of G, denoted $\beta(G)$. For each edge e in the cotree $G - T$, the unique simple cycle in the graph $T \cup \{e\}$ is called *the fundamental cycle* of e with respect to the spanning tree T.

The *deficiency* $\xi(G, T)$ of a spanning tree T for a graph G is defined to be the number of components of $G - T$ that have an odd number of edges. The *deficiency* $\xi(G)$ of the graph G is defined to be the minimum of $\xi(G, T)$ over all spanning trees T of G. A spanning tree T is a *Xuong tree* if $\xi(G, T) = \xi(G)$. Xuong [22] obtained a characterization of maximum genus $\gamma_M(G)$ of a graph G in terms of deficiency of the graph.

PROPOSITION 2.1 (see [22]). *Let G be a connected graph. Then*

$$\gamma_M(G) = \frac{\beta(G) - \xi(G)}{2}.$$

The effect of inserting or deleting an edge in a graph embedding has been discussed in the literature (see, e.g., [18, Theorem 1.2] and [10, section 2.4]). For the reader's convenience, we list the related results in the following propositions.

Let $\rho(G)$ be an embedding of a connected graph G. Suppose that we insert a new edge $e = [u, v]$ into the embedding $\rho(G)$, where u and v are vertices of G.

PROPOSITION 2.2. *If the edge-ends u and v of e are inserted between two corners of the same face F in $\rho(G)$, then the new edge e splits the face F into two faces with the embedding genus unchanged. If the edge-ends u and v of e are inserted between corners of two different faces F_1 and F_2 in $\rho(G)$, then both these faces are merged by e into one larger face with the embedding genus increased by* 1.

Figure 2.1 illustrates how an edge is inserted in the two different situations. Figure 2.1(a) inserts a new edge (u, x) between the two face corners $\langle vuw \rangle$ and $\langle wxv \rangle$, which belong to the same face. The insertion of the edge (u, x) splits the face $[vuwxv]$ into two faces $[uxvu]$ and $[uwxu]$ and does not change the embedding genus. Figure 2.1(b) inserts a new edge (u, y) between the two face corners $\langle vuw \rangle$ and $\langle wyx \rangle$, which belong to different faces. The insertion of the edge (u, y) merges the faces $[vuwxv]$ and $[wyxw]$ into a single face $[uwxvuyxwyu]$ and increases the embedding genus by 1.

The inverse operation of edge insertion is edge deletion. Let $\rho(H)$ be an embedding of a graph H and let e' be an edge in H such that e' is not a cut-edge. Then we have the following proposition.

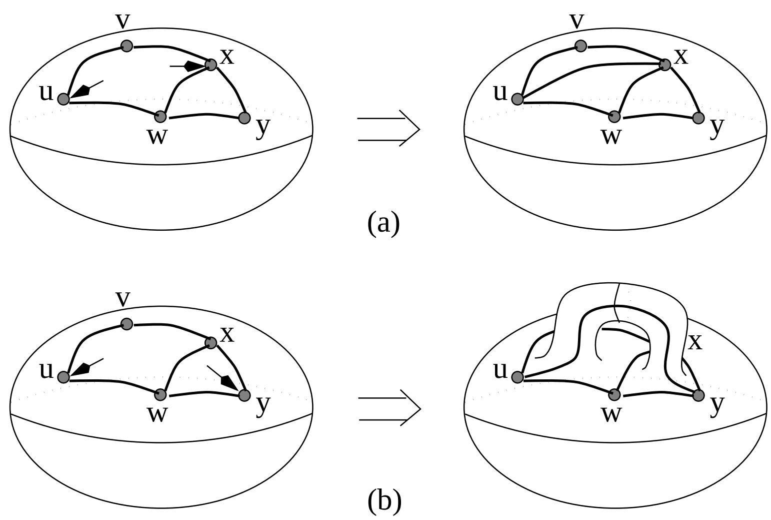


FIG. 2.1. (a) *inserting an edge between two corners of the same face;* (b) *inserting an edge between corners of two different faces.*

PROPOSITION 2.3. *If both sides of the edge* e' *belong to the same face* F *in* $\rho(H)$*, then deleting the edge* e' *from* $\rho(H)$ *splits the face* F *into two faces and decreases the embedding genus by* 1*; while if the edge* e' *is on the boundary of two different faces* F_1 *and* F_2 *in* $\rho(H)$*, then deleting the edge* e' *merges the two faces into a single face without changing the embedding genus.*

Note that inserting an edge into an embedding never decreases the embedding genus and that deleting an edge from an embedding never increases the embedding genus.

Following the standard definition [17], we define an *ear decomposition* $D = [P_1, P_2, \ldots, P_r]$ of a graph G to be a partition of edge set of G into an ordered collection of edge-disjoint simple paths $P_1, P_2, \ldots, P_r$ such that P_1 is a simple cycle and P_i, $i \geq 2$, is a path with only its endpoints in common with $P_1 + \cdots + P_{i-1}$. Each P_i is called an *ear*.

We now introduce the notions of pairing and maximum pairing in an ear decomposition.

DEFINITION 2.4. *A* pairing *in an ear decomposition* $D = [P_1, P_2, \ldots, P_r]$ *is a partition of the ears into pairs of* matched ears *and* single ears *such that each pair of matched ears consists of two consecutive ears* P_i *and* P_{i+1}*, where the ear* P_{i+1} *has an endpoint on the ear* P_i*. A* maximum pairing *of the ear decomposition* D *is a pairing that maximizes the number of pairs of matched ears.*

The following theorem shows that a maximum pairing of a given ear decomposition can be constructed very efficiently.

THEOREM 2.5. *Given an ear decomposition* $D = [P_1, P_2, \ldots, P_r]$ *of a* 2*-edge connected graph, a maximum pairing in* D *can be constructed in linear time.*

Proof. Represent the ear decomposition $D = [P_1, P_2, \ldots, P_r]$ as a graph H in which each ear P_i is a vertex. There is an edge between two vertices P_i and P_{i+1} in the graph H if and only if the ear P_{i+1} has an endpoint on the ear P_i. It is not hard to see that the graph H is a union of disjoint simple paths and a maximum matching in the graph H corresponds to a maximum pairing in the ear decomposition D. Finally, because of its special structure, a maximum matching in the graph H can be easily constructed in linear time. ∎

If a maximum pairing of an ear decomposition D has k pairs of matched ears, we say that the ear decomposition D is *k-paired*. A *maximum-paired ear decomposition* of a graph G is a k-paired ear decomposition such that k is the largest overall ear decomposition of G.

3. Ear decomposition and Xuong tree. In this section, we first develop efficient algorithms to show that constructing a maximum-paired ear decomposition of a graph and constructing a Xuong tree of the graph are linear-time equivalent. Based on these results, a conclusion is derived that a maximum-paired ear decomposition of a graph is k-paired if and only if the maximum genus of the graph is k.

Let H be a (not necessarily connected) graph. Two edges in the graph H are *adjacent* if they have an endpoint in common. An *adjacency matching* in H is a partition of edges of H into groups of one or two edges, called 1-*groups* and 2-*groups*, respectively, such that two edges in the same 2-group are adjacent. We say that the two edges in the same 2-group are *matched*, and the edge in a 1-group is *unmatched*. A *maximum adjacency matching* in H is an adjacency matching that maximizes the number of 2-groups.

LEMMA 3.1. *In a maximum adjacency matching of a graph H, the number of unmatched edges is equal to the number of components of H that have an odd number of edges. Moreover, a maximum adjacency matching of H can be constructed in linear time.*

Proof. The fact that the number of unmatched edges in a maximum adjacency matching of the graph H is equal to the number of components of H that have an odd number of edges was first observed by Xuong [22]. A linear-time algorithm that constructs a maximum adjacency matching of a graph has been given in [10, Theorem 3.6, pp. 529–530]. ∎

Let G be a 2-edge-connected graph and let T be a Xuong tree of G. We name each vertex in G by its preorder number in the tree T. Given a pair of vertices u and v, we denote by $lca(u, v)$ the least common ancestor of u and v in the tree T. We also say that a vertex w is the least common ancestor of a cotree edge $e = [u, v]$ if $w = lca(u, v)$. Note that the least common ancestor of a cotree edge e is the smallest vertex in the fundamental cycle of e.

Consider the algorithm *Xuong-to-Ears* in Figure 3.1. In the following, we prove that this algorithm constructs a maximum-paired ear decomposition for the graph G.

LEMMA 3.2. *If the graph G is 2-edge-connected, then every edge e in G is assigned a unique ear number $ear(e)$ in the algorithm Xuong-to-Ears.*

Proof. Let T be the Xuong tree in the algorithm *Xuong-to-Ears*. According to step 4 of the algorithm, each cotree edge e gets a different number $num(e)$. Therefore, each cotree edge e gets a unique ear number $ear(e)$ in step 5. For each tree edge e', since the graph G is 2-edge-connected, the edge e' must be in the fundamental cycle of some cotree edge. Therefore, the set $A_{e'}$ of the numbers $ear(e)$, where e is a cotree edge such that e' is in the fundamental cycle of e, is not empty. Now step 6 assigns the ear number $ear(e')$ to equal to the minimum in $A_{e'}$, which is unique since all cotree edges have different ear numbers. In consequence, for each tree edge e', the ear number $ear(e')$ is also well defined and unique. ∎

LEMMA 3.3. *The algorithm Xuong-to-Ears constructs a valid ear decomposition for the graph G. That is, there is an ear decomposition $D = [P_1, P_2, \ldots, P_r]$ of the graph G such that the ear P_i consists of exactly those edges that are assigned an ear number i by the algorithm Xuong-to-Ears.*

Algorithm *Xuong-to-Ears*

Input: a Xuong tree T of a 2-edge-connected graph G
Output: an ear decomposition of G

1. Rename each vertex of G by its preorder number in the tree T; construct the cotree $H = G - T$;
2. construct a maximum adjacency matching $\mathcal{M}$ in H;
3. sort all edges in the cotree H by their least common ancestors; let the sorted list be L_1;
4. assign each cotree edge e a number $num(e)$ as follows: for each pair of matched edges $L_1(i)$ and $L_1(j)$ in $\mathcal{M}$, where $i < j$, assign $num(L_1(i)) = 2i$ and $num(L_1(j)) = 2i + 1$; for each unmatched edge $L_1(k)$ in $\mathcal{M}$, assign $num(L_1(k)) = 2k$;
5. sort all cotree edges by their new assigned numbers $num(\cdot)$; let the sorted list be L_2; now for each cotree edge $e = L_2(i)$, assign $ear(e) = i$;
6. for each tree edge e', assign $ear(e') = \min\{ear(e)\}$, where the minimum is taken over all cotree edges e whose fundamental cycle contains e'.

FIG. 3.1. *The algorithm Xuong-to-Ears.*

Proof. Let the cotree edges in the sorted list L_2 be $e_1, e_2, \ldots, e_r$, with $ear(e_i) = i$ for $i = 1, 2, \ldots, r$. First note that by our construction, only edges in the fundamental cycle of the cotree edge e_i may be assigned an ear number i, and that if an edge in the fundamental cycle of e_i has an ear number different from i, then the edge must have an ear number less than i. In particular, the fundamental cycle of e_1 contains the root of the tree T and every edge in it is assigned an ear number 1. This forms the first ear, P_1.

Consider the cotree edge $e_i = [x_i, y_i]$, $i > 1$, with $ear(e_i) = i$. Let the directed tree paths (directed from child to parent) from x_i and y_i to the vertex $lca(x_i, y_i)$ be Q_x and Q_y, respectively. If all the tree edges on Q_x and Q_y are assigned ear number i, then the ear P_i is the fundamental cycle of e_i. Note that if the vertex $lca(x_i, y_i)$ is not the root of T, then it must be contained in the fundamental cycle of another cotree edge e_j, with $j < i$.

Suppose that some edge in the fundamental cycle of e_i has an ear number less than i. We show that there are two vertices v_x and v_y on the paths Q_x and Q_y, respectively, such that all edges before the vertex v_x (respectively, v_y) on the path Q_x (respectively, Q_y) are assigned ear number i and all edges after the vertex v_x (respectively, v_y) on the path Q_x (respectively, Q_y) are assigned ear number less than i. Let f_1 be the first edge on Q_x with $ear(f_1) = j < i$. Consider the cotree edge $e_j = [x_j, y_j]$ with $ear(e_j) = j$. According to steps 4 and 5 of the algorithm, we must have either $lca(e_j) \leq lca(e_i)$, or $lca(e_j) > lca(e_i)$ and $\{e_{j-1}, e_j\}$ is a matched pair in $\mathcal{M}$ with $lca(e_{j-1}) \leq lca(e_i)$, for the cotree edge $e_{j-1} = [x_{j-1}, y_{j-1}]$. We first observe that because there is a unique directed tree path from the edge f_1 to the root of the tree and the edge f_1 is shared by the fundamental cycles of the cotree edges e_j and e_i, the vertex $lca(e_j)$ is either an ancestor or a descendent of the vertex $lca(e_i)$.

Suppose $lca(e_j) \leq lca(e_i)$. Then $lca(e_j)$ is an ancestor of $lca(e_i)$. Thus, all edges on the path Q_x after the edge f_1 belong to the fundamental cycle of the cotree edge e_j and should have ear numbers of at most j, which is smaller than i.

On the other hand, suppose $lca(e_j) > lca(e_i)$ and $\{e_{j-1}, e_j\}$ is a matched pair in $\mathcal{M}$ with $lca(e_{j-1}) \leq lca(e_i)$ for the cotree edge $e_{j-1} = [x_{j-1}, y_{j-1}]$; then $lca(e_j)$ is on the path Q_x. Since the partial path on Q_x from the edge f_1 to the vertex $lca(e_j)$ belongs to the fundamental cycle of the cotree edge e_j, the edges on this partial path of Q_x have ear numbers of at most j, which is smaller than i. It remains to examine the partial path of Q_x from the vertex $lca(e_j)$ to the vertex $lca(e_i)$. Since the cotree edges e_j and e_{j-1} are matched (thus they share a common endpoint), the least common ancestors $lca(e_j)$ and $lca(e_{j-1})$ must be on the same tree path from this endpoint to the root of the tree. Now since $lca(e_{j-1}) \leq lca(e_i) < lca(e_j)$ and the vertex $lca(e_i)$ is on the tree path from $lca(e_j)$ to the root of the tree, we conclude that the partial path of Q_x from the vertex $lca(e_j)$ to the vertex $lca(e_i)$ must be entirely contained in the fundamental cycle of the cotree edge e_{j-1}. Thus, all edges on this partial path of Q_x have ear numbers of at most $j-1$, which is smaller than i. This concludes that all edges on the path Q_x after the edge f_1 have ear numbers smaller than i.

Completely similar proof shows that if we let f_2 be the first edge with an ear number less than i on the path Q_y, then all edges on the path Q_y after the edge f_2 should have ear numbers smaller than i.

Therefore, we can find a vertex v_x on the path Q_x and a vertex v_y on the path Q_y such that the edges on the path between the vertices v_x and v_y that contains the edge e_i in the fundamental cycle of e_i are the only edges that have ear number i. These edges form the ear P_i. □

Let $H = G - T$ be the cotree and let $\mathcal{M}$ be the maximum adjacency matching in H constructed by the algorithm *Xuong-to-Ears* in step 2. By Lemma 3.1, the number of unmatched edges in $\mathcal{M}$ is equal to the deficiency $\xi(G)$ of the graph G. Thus, the number of pairs of matched edges in $\mathcal{M}$ is $(\beta(G) - \xi(G))/2$, which is equal to $\gamma_M(G)$ by Proposition 2.1. Since each pair of matched edges in $\mathcal{M}$ is adjacent in the list L_2, their ear numbers differ by exactly 1. Thus, their corresponding ears are consecutive in the ear decomposition. Moreover, they have at least one vertex in common. This gives us the following lemma immediately.

LEMMA 3.4. *The ear decomposition constructed by the algorithm Xuong-to-Ears is k-paired, where $k \geq \gamma_M(G)$.*

In fact, the integer k in Lemma 3.4 cannot be larger than $\gamma_M(G)$ because of the following lemma.

LEMMA 3.5. *If a graph G has a k-paired ear decomposition, then $\gamma_M(G) \geq k$.*

Proof. Let $D = [P_1, P_2, \ldots, P_r]$ be a k-paired ear decomposition of the graph G and let $\mathcal{P}$ be a maximum pairing of D. We construct an embedding of G by first embedding the cycle P_1 into the plane then inserting the ears $P_2, \ldots, P_r$ one by one in that order into the embedding. Inductively, suppose that we have embedded the ears $P_1, \ldots, P_{j-1}$ and the next two ears P_j and P_{j+1} are a pair of matched ears. We first arbitrarily insert the ear P_j. If this increases the embedding genus, then we arbitrarily insert the ear P_{j+1}. On the other hand, if inserting the ear P_j does not increase the embedding genus, then by Proposition 2.2 P_j must split a face F of the embedding of $P_1 + P_2 + \cdots + P_{j-1}$ into two faces F_1 and F_2. Therefore, all edges of the ear P_j are on the boundary of the two faces F_1 and F_2. Now since the ear P_{j+1} is matched with P_j, it has an endpoint on the ear P_j. Thus, no matter where the other endpoint of P_{j+1} is (including the case that the two endpoints of P_{j+1} are attached to the same vertex), we are always able to insert the ear P_{j+1} so that the two endpoints of the ear P_{j+1} are inserted into corners of two different faces of the embedding of $P_1 + P_2 + \cdots + P_{j-1} + P_j$. By Proposition 2.2 again, this increases the embedding genus by 1.

We also arbitrarily insert all the single ears.

According to the above description, we can always insert each pair of matched ears in $\mathcal{P}$ in such a way that increases the embedding genus by at least 1; therefore, this construction will result in an embedding of genus at least k. □

Now we are ready for our first theorem in this section.

THEOREM 3.6. *A 2-edge-connected graph G has maximum genus k if and only if every maximum-paired ear decomposition of G is k-paired. Moreover, a maximum-paired ear decomposition of the graph G can be constructed from a Xuong tree of G in linear time.*

Proof. Lemma 3.5 shows that the graph G does not have a k-paired ear decomposition, with $k > \gamma_M(G)$, while Lemma 3.4 and Lemma 3.5 together show that the algorithm *Xuong-to-Ears* constructs a $\gamma_M(G)$-paired ear decomposition for the graph G. To complete the proof of the theorem, we need to show only that the algorithm *Xuong-to-Ears* runs in linear time.

Step 1 of the algorithm can be done trivially in linear time. Step 2 can be done in linear time according to Lemma 3.1. To compute the least common ancestor for each cotree edge, we use Schieber and Vishkin's algorithm [19], which can compute the least common ancestor of any two vertices in constant time with a linear-time preprocessing. All sortings in the algorithm can be implemented using bucket sorting that takes linear time. Finally, to assign ear numbers to the tree edges, we pick the cotree edges sorted in the list L_2. For each cotree edge $e_i = [x_i, y_i]$, we traverse the two tree paths Q_x and Q_y from the vertices x_i and y_i, respectively, to the least common ancestor of e_i and stop at a vertex that belongs to an ear of smaller index. By the proof of Lemma 3.3, exactly those edges traversed in the process should be assigned the ear number $ear(e_i)$. This process obviously can be done in linear time. This proves that the algorithm *Xuong-to-Ears* runs in linear time. □

Furst, Gross, and McGeoch [10] have developed an $O(m^2 n \log^6 n)$-time algorithm that constructs a Xuong tree for a graph G, where m is the number of edges and n is the number of vertices in G. This result with Theorem 3.6 gives us a polynomial-time algorithm for constructing a maximum-paired ear decomposition given a 2-edge-connected graph.

THEOREM 3.7. *A maximum-paired ear decomposition of a 2-edge-connected graph G of n vertices and m edges can be constructed in time $O(m^2 n \log^6 n)$.*

Theorem 3.6 indicates that a maximum-paired ear decomposition of a 2-edge-connected graph can be constructed from a Xuong tree of the graph. In the following, we show the converse of this fact, that is, how a Xuong tree of a 2-edge-connected graph is constructed from a maximum-paired ear decomposition of the graph.

LEMMA 3.8. *Let $D = [P_1, P_2, \ldots, P_r]$ be an ear decomposition of a graph G. Let S be a subset of ears in D. Let G_S be a subgraph of G obtained by deleting one edge from each ear in S. Then the graph G_S is a connected spanning subgraph of G.*

Proof. It suffices to prove the lemma for the case that S contains all ears in D.

Since no vertices are deleted, the graph G_S is a spanning subgraph of the graph G. The connectedness of the graph G_S can be proved easily by induction on the number of ears in the ear decomposition D, based on the following observation: Suppose that the graph $G'_S = (P_1 + \cdots + P_{r-1}) - S$ is connected; then every edge in the ear P_r is in a cycle in the graph $G'_S + P_r$. Thus, deleting any edge in P_r from $G'_S + P_r$ still leaves a connected graph. □

Now consider the algorithm *Ears-to-Xuong* in Figure 3.2.

THEOREM 3.9. *Given a maximum-paired ear decomposition of a 2-edge-connected graph, the algorithm Ears-to-Xuong constructs a Xuong tree of the graph G in linear time.*

Proof. By Theorem 2.5, step 1 can be done in linear time. By Theorem 3.6, the pairing $\mathcal{P}$ has $\gamma_M(G)$ pairs of matched ears.

Algorithm *Ears-to-Xuong*

Input: a maximum-paired ear decomposition D of G
Output: a Xuong tree T of the graph G

1. Construct a maximum pairing $\mathcal{P}$ in the ear decomposition D; let the $\gamma_M(G)$ pairs of matched ears in $\mathcal{P}$ be $\{P_{i_j}, P_{i_j+1}\}$, $1 \le j \le \gamma_M(G)$;
2. for each pair $\{P_{i_j}, P_{i_j+1}\}$, let v_j be a vertex shared by P_{i_j} and P_{i_j+1}, delete an edge e_{i_j} in P_{i_j} and an edge e_{i_j+1} in P_{i_j+1} such that both edges are incident on v_j;
3. let G_S be the subgraph of G after the edge deletions in step 2, construct a spanning tree T in G_S; T is a Xuong tree of the graph G.

FIG. 3.2. *The algorithm Ears-to-Xuong.*

It is easy to see that steps 2 and 3 take linear time. By Lemma 3.8, the graph G_S is a connected spanning subgraph of G. Therefore, the spanning tree T of G_S is also a spanning tree of the graph G. Moreover, since in the cotree $G-T$ there are $\gamma_M(G)$ pairs of adjacent edges e_{i_j} and e_{i_j+1}, $1 \le j \le \gamma_M(G)$, the number of components in $G-T$ that has an odd number of edges cannot be larger than $\beta(G) - 2\gamma_M(G) = \xi(G)$. Therefore, the spanning tree T is a Xuong tree of the graph G. □

Combining Theorem 3.6 and Theorem 3.9, we obtain the next theorem.

THEOREM 3.10. *Constructing a Xuong tree of a* 2*-edge-connected graph and constructing a maximum-paired ear decomposition of the graph are linear-time equivalent.*

In a pairing of an ear decomposition D, pairs of matched ears and single ears are, in general, interlaced. The *canonical pairing* of D is the pairing of D that pairs the first $2k$ ears in the ear decomposition D into k pairs of matched ears and lets all other ears be single ears, with k being the largest possible integer. Note that the number of pairs of matched ears in the canonical pairing of D may be less than the number of pairs of matched ears in a maximum pairing of D.

The canonical pairing of an ear decomposition can be trivially constructed, and it has a more uniform and simpler structure than a general pairing in the ear decomposition. More importantly, the canonical pairing of ear decompositions has found applications in the study of upper-embeddable subgraphs (see Theorem 5.2) and in the study of lower bound on the maximum genus and the average genus of a graph [13]. These results are based on the following theorem, which shows that Theorem 3.6 is still valid even if we restrict it to only canonical pairings of ear decompositions.

THEOREM 3.11. *Every* 2*-edge-connected graph G has an ear decomposition whose canonical pairing has $\gamma_M(G)$ pairs of matched ears.*

Proof. Let $D = [P_1, P_2, \ldots, P_r]$ be an arbitrary maximum-paired ear decomposition of the 2-edge-connected graph G. Let $\mathcal{P}$ be a maximum pairing of D. By Theorem 3.6, $\mathcal{P}$ has $\gamma_M(G)$ pairs of matched ears.

If the canonical pairing of D has less than $\gamma_M(G)$ pairs of matched ears, then we can find an index i such that in the pairing $\mathcal{P}$, the ear P_i is a single ear, while the ears P_{i+1} and P_{i+2} are a pair of matched ears. We show that we can always "shift" the single ear P_i two positions to the right and the pair of matched ears P_{i+1} and P_{i+2} one position to the left. There are three possible cases.

Case 1. No endpoint of the ears P_{i+1} and P_{i+2} is an interior vertex of the ear P_i. Then the sequence

$$D_1 = [P_1, \ldots, P_{i-1}, P_{i+1}, P_{i+2}, P_i, P_{i+3}, \ldots, P_r]$$

is also a valid maximum-paired ear decomposition of the graph G. Note that the single ear P_i has been shifted two positions to the right and the pair of matched ears P_{i+1} and P_{i+2} has been shifted one position to the left.

Case 2. The ear P_{i+1} has an endpoint on an interior vertex of the ear P_i. Then we make the ears P_i and P_{i+1} a pair of matched ears and let P_{i+2} be a single ear. Thus, in this case, we also have shifted a single ear in the pairing $\mathcal{P}$ two positions to the right and shifted a pair of matched ears one position to the left.

Case 3. The ear P_{i+2} has an endpoint on an interior vertex of P_i. Let the endpoint of P_{i+2} on P_{i+1} be u, and let the ear P_{i+1} be the path $P_{i+1}^{(1)}uP_{i+1}^{(2)}$, where $P_{i+1}^{(1)}$ and $P_{i+1}^{(2)}$ are subpaths on P_{i+1}. Without loss of generality, we assume $P_{i+1}^{(1)}$ contains at least one edge. Then we rearrange the ears P_i, P_{i+1}, and P_{i+2} into three new ears P_i', P_{i+1}', and P_{i+2}', as follows: $P_i' = P_i$, $P_{i+1}' = P_{i+2} + P_{i+1}^{(2)}$, and $P_{i+2}' = P_{i+1}^{(1)}$. It is easy to verify that the sequence

$$D_2 = [P_1, \dots, P_{i-1}, P_i', P_{i+1}', P_{i+2}', P_{i+3}, \dots, P_r]$$

is a valid maximum-paired ear decomposition of the graph G. If we construct a pairing of D_2 that is identical to $\mathcal{P}$ except that we make P_i' and P_{i+1}' a pair of matched ears and make P_{i+2}' a single ear, then again this pairing has "shifted" in the pairing $\mathcal{P}$ a single ear two positions to the right and a pair of matched ears one position to the left.

Therefore, for any maximum-paired ear decomposition D of the graph G, if the canonical pairing has less than $\gamma_M(G)$ pairs of matched ears, then we can start with a maximum pairing $\mathcal{P}$ of D and apply the above process that constructs a maximum-paired ear decomposition D' of G and a maximum pairing $\mathcal{P}'$ of D' that shifts in the pairing $\mathcal{P}$ a single ear two positions to the right and a pair of matched ears one position to the left. If the canonical pairing of the resulting ear decomposition D' still has less than $\gamma_M(G)$ pairs of matched ears, then we apply the above process on D' and $\mathcal{P}'$ again. It is easy to see that after a finite number of applications of the process, we must end up with a maximum-paired ear decomposition of the graph G whose canonical pairing has exactly $\gamma_M(G)$ pairs of matched ears. □

We point out that, with a careful implementation of the process presented in the proof of Theorem 3.11, we can derive a linear-time algorithm that, given a maximum-paired ear decomposition of a graph G, constructs a maximum-paired ear decomposition of G whose canonical pairing has $\gamma_M(G)$ pairs of matched ears. We leave the details of this implementation to the interested reader.

4. Maximum genus embedding and Xuong tree. We have shown that constructing a maximum-paired ear decomposition and constructing a Xuong tree are linear-time equivalent. It is well known that the Xuong tree of a graph is closely related to the maximum genus embedding of the graph [10, 22], which induces a close relationship between maximum-paired ear decompositions and maximum genus embeddings. In this section, we discuss the algorithmic relationship between constructing a Xuong tree of a graph and constructing a maximum genus embedding of the graph.

An $O(m^2)$-time algorithm was developed by Furst, Gross, and McGeoch [10], based on the construction by Xuong [22], that constructs a maximum genus embedding of a graph G of m edges from a Xuong tree of the graph. The basic idea of the algorithm is to start with a one-face embedding of the Xuong tree T, for which the cotree $G - T$ has $\gamma_M(G)$ pairs of matched edges, then add each pair of the matched edges in such a way that increases the embedding genus by 1. This construction results in a maximum genus embedding of the graph G. Recently, Chen [2] developed a new

Algorithm *Embed-to-Xuong*

Input: a maximum genus embedding $\rho(G)$ of G
Output: a Xuong tree T of G

1. $G' = G$; $\rho'(G') = \rho(G)$;
2. **while** there is an edge e on boundary of two different faces in $\rho'(G')$
   delete the edge e from G' and $\rho'(G')$;
3. let $M = \phi$;
4. **while** G' is not empty **do**
   3.1. find an edge e_1 in G' such that a subwalk δ_1 between the two sides of e_1 is the shortest in the face boundary walk of $\rho'(G')$;
   3.2. **if** δ_1 is empty
   then delete e_1 and its degree-1 end from $\rho'(G')$ and G'
   else let $\delta_1 = l(e_2)\delta_1'$; delete e_1 and e_2 from $\rho'(G')$ and G'; add e_1 and e_2 to the set M;
5. let $G_0 = G - M$; construct a spanning tree T for G_0, which is a Xuong tree for the graph G.

FIG. 4.1. *The algorithm Embed-to-Xuong.*

data structure for graph embeddings on which each of the basic embedding operations such as edge insertion and edge deletion can be performed in time $O(\log m)$. Based on this new data structure, the above construction from Xuong tree to maximum genus embedding can be implemented in an algorithm of running time $O(m \log m)$.

PROPOSITION 4.1 (see [2, 10]). *A maximum genus embedding of a graph can be constructed from a Xuong tree of the graph in time $O(m \log m)$.*

We now consider how to construct a Xuong tree of a graph G from a maximum genus embedding $\rho(G)$ of the graph. This is done in two steps. First, we find a spanning subgraph G' of G such that the induced embedding $\rho'(G')$ of G' from $\rho(G)$ is of genus $\gamma_M(G)$ and has only one face. Second, we construct a Xuong tree of G from the embedding $\rho'(G')$. Consider the algorithm *Embed-to-Xuong* in Figure 4.1.

THEOREM 4.2. *Given a maximum genus embedding of a graph of m edges, the algorithm Embed-to-Xuong constructs a Xuong tree of the graph in time $O(m^2)$.*

Proof. According to Proposition 2.3, deleting an edge on the boundary of two different faces in an embedding reduces the number of faces by 1 and does not change the embedding genus. Moreover, note that a cut-edge of a graph always has its two sides on the boundary of the same face in any embedding of the graph. Therefore, deleting an edge on the boundary of two different faces in an embedding does not disconnect the graph. Consequently, step 2 results in a connected subgraph G' of G and a one-face embedding $\rho'(G')$ of genus $\gamma_M(G)$.

Now we consider step 4. The method we use here is similar to the one used by Gross and Tucker in establishing Xuong's characterization of graph maximum genus (see [12, proof of Lemma 3.4.9]). Each time, we look at the subgraph G' and the one-face embedding $\rho'(G')$. We either delete a single edge from $\rho'(G')$ without changing the embedding genus, or delete a pair of adjacent edges from $\rho'(G')$ and decrease the embedding genus by 1.

Consider the boundary walk of the unique face of $\rho'(G')$. Suppose that e_1 is an edge such that a subwalk between the two sides $l(e_1)$ and $r(e_1)$ of e_1 is the shortest. Therefore, the boundary walk of the face of $\rho'(G')$ can be written as $l(e_1)\delta_1 r(e_1)\delta_2$, where δ_1 and δ_2 are subwalks, and δ_1 is the shortest in the above sense. There are two cases.

If the subwalk δ_1 is empty, then the boundary walk of the face is $l(e_1)r(e_1)\delta_2$, and one end of the edge e_1 must have degree 1. Thus, deleting the edge e_1 and its degree-1 end from the embedding $\rho'(G')$ neither changes the embedding genus, nor disconnects the graph G'. Moreover, the embedding after deleting the edge e_1 is still a one-face embedding.

If the subwalk δ_1 is not empty, then the boundary walk of the face can be written as $l(e_1)l(e_2)\delta_1' r(e_1)\delta_2$. The other side $r(e_2)$ of the edge e_2 should not be contained in the subwalk δ_1', since otherwise the subwalk between the two sides of e_2 would be shorter than δ_1, contradicting the selection of the edge e_1. Therefore, the boundary walk of the face of the embedding $\rho'(G')$ can be further written as $l(e_1)l(e_2)\delta_1' r(e_1)\delta_2' r(e_2)\delta_2''$. Note that the edges e_1 and e_2 are adjacent. We claim that deleting the pair of adjacent edges e_1 and e_2 reduces the embedding genus by 1 and does not disconnect the graph G'.

First consider the graph $G' - \{e_1\}$. All edges that have an edge side in the subwalk $l(e_2)\delta_1'$ are contained in the same connected component of $G' - \{e_1\}$, and all edges that have an edge side in the subwalk $\delta_2' r(e_2)\delta_2''$ are contained in the same connected component of $G' - \{e_1\}$. Since the edge e_2 is in both components, we conclude that the graph $G' - \{e_1\}$ is connected. By Proposition 2.3, deleting the edge e_1 splits the unique face $l(e_1)l(e_2)\delta_1' r(e_1)\delta_2' r(e_2)\delta_2''$ of $\rho'(G')$ into two faces, $l(e_2)\delta_1'$ and $\delta_2' r(e_2)\delta_2''$, and reduces the embedding genus by 1. Now since the edge e_2 is on the boundary of these two different faces, the edge e_2 is not a cut-edge in the graph $G' - \{e_1\}$, thus deleting the edge e_2 still results in a connected graph. Moreover, by Proposition 2.3 again, deleting the edge e_2 merges the two faces into a single face without changing the embedding genus. Thus, after deleting the edges e_1 and e_2, the graph G' remains connected and $\rho'(G')$ is a one-face embedding of G' with embedding genus decreased by exactly 1, which prepares for the next execution of the **while** loop body in step 4.

Since each deletion of a pair of adjacent edges in step 4 reduces the embedding genus by exactly 1, we conclude that after step 4, the set M contains exactly $\gamma_M(G)$ pairs of adjacent edges.

We need to ensure that the subgraph $G_0 = G - M$ in step 5 is connected. This can be proved by induction on the number of times the **while** loop body in step 4 adds a pair of adjacent edges to the set M. Before any pair of adjacent edges is added to M, $M = \phi$ and $G - M$ is a connected graph. Now suppose that a new pair (e_1, e_2) of adjacent edges is to be added to the set M. By the inductive hypothesis, before adding (e_1, e_2) to M, the graph $G - M$ is connected. As we proved above, the edge e_1 is not a cut-edge of the subgraph G' and the edge e_2 is not a cut-edge of the subgraph $G' - \{e_1\}$. Therefore, the edge e_1 is contained in a cycle in G' and the edge e_2 is contained in a cycle in $G' - \{e_1\}$. Note that the graph G' is actually a subgraph of $G - M$. Thus, the edge e_1 is contained in a cycle in $G - M$ and the edge e_2 is contained in a cycle in $(G - M) - \{e_1\}$. Therefore, the graph $(G - M) - \{e_1, e_2\}$ is connected. In consequence, after adding the adjacent pair (e_1, e_2) to the set M, the graph $G - M$ is still connected.

Since no vertex is removed, the connected subgraph $G_0 = G - M$ in step 5 is also a spanning subgraph of G. Therefore, the tree T constructed by the algorithm *Embed-to-Xuong* is a spanning tree of the graph G. Moreover, since the set M contains $\gamma_M(G)$ pairs of adjacent edges, the cotree $G - T$ contains at least $\gamma_M(G)$ pairs of adjacent edges. By the definition of a Xuong tree, the tree T constructed by the algorithm *Embed-to-Xuong* is a Xuong tree.

We analyze the algorithm. Each execution of the **while** loop body in step 2 takes time $O(m)$ by traversing the face boundaries of the embedding $\rho'(G')$. Thus, step 2 takes time $O(m^2)$. For each execution of the **while** loop body in step 4, we find an edge e_1 in the one-face embedding $\rho'(G')$ such that a subwalk between the two sides of e_1 is the shortest. This can be done by traversing the face boundaries and recording the (circular) position for each edge side. Then the length of the subwalks between two edge sides of an edge can be computed in constant time. Thus, finding the edge e_1 takes time $O(m)$. All other steps can also be done in time $O(m)$. Therefore, step 4 of the algorithm takes time $O(m^2)$. In conclusion, the time complexity of the algorithm *Embed-to-Xuong* is bounded by $O(m^2)$. □

5. Conclusions. Combining Theorem 3.10, Proposition 4.1, and Theorem 4.2, we obtain the following interesting theorem.

THEOREM 5.1. *Constructing a maximum-paired ear decomposition of a 2-edge-connected graph G and constructing a maximum genus embedding of the graph are polynomial-time equivalent. More precisely, a maximum genus embedding of G can be constructed from a maximum-paired ear decomposition of G in time $O(m \log m)$, and a maximum-paired ear decomposition of G can be constructed from a maximum genus embedding of G in time $O(m^2)$.*

Theorem 5.1 should have further applications in the study of theory and complexity of graph embeddings and graph connectivities. We illustrate two applications of the theorem as follows.

A graph G is *upper-embeddable* if its maximum genus is equal to $\lfloor \beta(G)/2 \rfloor$. Upper-embeddability of graphs has been studied extensively. Many interesting graph classes, including complete graphs and complete bipartite graphs, are upper-embeddable (see [18]). The following theorem relates the upper-embeddability to the connectivity of a 2-edge-connected graph.

THEOREM 5.2. *Every 2-edge-connected graph G contains an upper-embeddable subgraph G' such that G' is 2-edge-connected and the maximum genus of G' is $\gamma_M(G)$. Moreover, the subgraph G' can be constructed from G in time $O(m^2 n \log^6 n)$.*

Proof. We first construct in time $O(m^2 n \log^6 n)$ a maximum genus embedding $\rho(G)$ of the graph G using the algorithm developed in [10]. According to Theorem 4.2, a Xuong tree T of G can be constructed from $\rho(G)$ in time $O(m^2)$. Now based on the Xuong tree T, we construct a maximum-paired ear decomposition D of the graph G, using Algorithm *Xuong-to-Ears.* Finally, based on the maximum-paired ear decomposition D, we construct an ear decomposition D_0 whose canonical pairing has $\gamma_M(G)$ pairs of matched ears (Theorem 3.11 and its proof). Let the first $2\gamma_M(G)$ ears in D_0 be $P_1, P'_1, \ldots, P_t, P'_t$, where $t = \gamma_M(G)$, and ears P_i and P'_i are matched, $i = 1, \ldots, t$. It is easy to verify that the subgraph $G' = P_1 + P'_1 + \cdots + P_t + P'_t$ of G is upper-embeddable, has maximum genus $\gamma_M(G)$, and is 2-edge connected. □

A more complicated application of Theorem 5.1 is given by Kanchi and Chen [13], in which, based on a maximum-paired ear decomposition of 2-edge-connected graphs, a tight lower bound for maximum genus of 2-edge-connected graphs is derived (see also [3]).

We believe that Theorem 5.1 may suggest an alternative and more direct approach for developing polynomial-time algorithms for maximum genus embeddings of a graph.

Acknowledgments. The authors would like to thank Jonathan Gross, Arkady Kanevsky, Vijaya Ramachandran, and Arthur White for their comments and discussion. The authors also express their sincere thanks to two anonymous referees, whose comments and suggestions have significantly improved the presentation of the paper.

REFERENCES

[1] J. Chen, *A linear-time algorithm for isomorphism of graphs of bounded average genus*, SIAM J. Discrete Math., 7 (1994), pp. 614–631.

[2] J. Chen, *Algorithmic graph embeddings*, Theoret. Comput. Sci., 181 (1997), pp. 247–266.

[3] J. Chen, D. Archdeacon, and J. L. Gross, *Maximum genus and connectivity*, Discrete Math., 149 (1996), pp. 19–29.

[4] J. Chen and J. L. Gross, *Limit points for average genus* (I): 3*-connected and* 2*-connected simplicial graphs*, J. Combin. Theory Ser. B, 55 (1992), pp. 83–103.

[5] J. Chen and J. L. Gross, *Kuratowski-type theorems for average genus*, J. Combin. Theory Ser. B, 57 (1993), pp. 100–211.

[6] J. Chen, J. L. Gross, and R. G. Rieper, *Overlap matrices and imbedding distributions*, Discrete Math., 128 (1994), pp. 73–94.

[7] J. Chen and A. Kanevsky, *On assembly of four-connected graphs*, in Graph-Theoretic Concepts in Computer Science, Lecture Notes in Comput. Sci. 657, Springer-Verlag, New York, 1993, pp. 158–169.

[8] J. Chen, A. Kanevsky, and R. Tamassia, *Algorithm for triconnected ear decomposition of a graph*, presented at 22nd Southeastern International Conference on Combinatorics, Graph Theory, and Computing, Baton Rouge, LA, 1991.

[9] 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.

[10] M. Furst, J. L. Gross, and L. A. McGeoch, *Finding a maximum-genus graph imbedding*, J. Assoc. Comput. Mach., 35 (1988), pp. 523–534.

[11] H. N. Gabow and M. Stallmann, *Efficient algorithms for graphic matroid intersection and parity*, in Automata, Languages and Programming, Lecture Notes in Comput. Sci. 194, Springer-Verlag, New York, 1985, pp. 210–220.

[12] J. L. Gross and T. W. Tucker, *Topological Graph Theory*, Wiley-Interscience, New York, 1987.

[13] S. P. Kanchi and J. Chen, *Tight lower bound on maximum genus of a* 2*-connected simplicial graph*, submitted.

[14] A. Kanevsky and V. Ramachandran, *Improved algorithms for graph four-connectivity*, J. Comput. System Sci., 42 (1991), pp. 288–306.

[15] L. Lovasz, *Computing ears and branchings in parallel*, in Proc. 26th Annual IEEE Symposium on Foundations of Computer Science, Portland, OR, IEEE Computer Society Press, Los Alamitos, CA, 1985, pp. 464–467.

[16] E. Nordhaus, B. Stewart, and A. White, *On the maximum genus of a graph*, J. Combin. Theory Ser. B, 11 (1971), pp. 258–267.

[17] V. Ramachandran, *Parallel open ear decomposition with applications to graph biconnectivity and triconnectivity*, in Synthesis of Parallel Algorithms, J. Reif, ed., Morgan-Kaufmann, 1993, pp. 275-340.

[18] R. Ringeisen, *Survey of results on the maximum genus of a graph*, J. Graph Theory, 3 (1979), pp. 1–13.

[19] B. Schieber and U. Vishkin, *On finding lowest common ancestors: Simplification and parallelization*, SIAM J. Comput., 17 (1988), pp. 1253–1262.

[20] P. J. Slater, *A classification of* 4*-connected graphs*, J. Combin. Theory Ser. B, 17 (1974), pp. 281–298.

[21] H. Whitney, *Non-separable and planar graphs*, Trans. Amer. Math. Soc., 34 (1932), pp. 339–362.

[22] N. H. Xuong, *How to determine the maximum genus of a graph*, J. Combin. Theory Ser. B, 26 (1979), pp. 217–225.

SIAM J. DISCRETE MATH.
Vol. 12, No. 2, pp. 243–251

NEW BOUNDS ON COVERING RADIUS AS A FUNCTION OF DUAL DISTANCE*

TERO LAIHONEN† AND SIMON LITSYN‡

Abstract. In this paper we estimate covering radius when dual distance is known. We derive new bounds on covering radii of linear codes. A bound for self-complementary codes is also presented. The improvements of these bounds on the known results are based on the knowledge of the cardinality of constant weight codes and on the behavior of Hahn polynomials and discrete Chebyshev polynomials.

Key words. covering radius, dual distance, Hahn polynomials, discrete Chebyshev polynomials

AMS subject classifications. 94B65, 94B75, 05B40

PII. S0895480197331703

1. Introduction.

Covering radius is a fundamental geometric parameter of a code, and it has applications for instance to problems of data compression and write-once memories [1].

The problem of upperbounding the covering radius has been an object of growing interest for a few decades. In order to estimate covering radius we may profit from the natural parameter called dual distance. It is often known or it can be at least estimated from the construction of the code (for a linear code it is the minimum distance of the dual code).

Ever since the seminal articles by Delsarte [3] and Helleseth, Kløve, and Mykkeltveit [6] there have been many papers dealing with bounds on covering radius as a function of dual distance (see, e.g., [1, 4, 7, 12, 15, 16, 19, 9]).

Specifically, in 1990 Tietäväinen proved the following remarkable results [19, 20]:

1) Let $\mathcal{C} = (C_n)_{n=1}^{\infty}$ be a sequence of binary codes C_n of length n, dual distance $d' = d'(n)$, and covering radius $R = R(n)$, where $d'/n \to \delta'$ and $R/n \to \rho$ when $n \to \infty$. Then

$$\rho \leq \frac{1}{2}(1 - \sqrt{\delta'(2-\delta')}). \tag{1.1}$$

2) Let $0 < \delta' < 1/2$. There are sequences $\mathcal{C}$ such that

$$\rho \geq H_2^{-1}(1 - H_2(\delta')),$$

where $H_2(x) = -x \log_2 x - (1-x)\log_2(1-x)$ is the binary entropy function.

The upper bound was generalized by Fazekas and Levenshtein [4] to other metric spaces.

In 1996 Litsyn and Tietäväinen [12] introduced a new method for binary linear codes which generalizes the approach presented in [5] and [18]. Using this method and (regular) Chebyshev polynomials they were able to improve on the previous bounds for relatively large values of dual distance.

*Received by the editors December 17, 1997; accepted for publication (in revised form) September 14, 1998; published electronically April 29, 1999.

http://www.siam.org/journals/sidma/12-2/33170.html

†Department of Mathematics, University of Turku, FIN-20014 Turku, Finland (terolai@utu.fi).
The research of this author was supported by the Academy of Finland.

‡Department of Electrical Engineering-Systems, Tel Aviv University, Ramat-Aviv, 69978, Israel (litsyn@eng.tau.ac.il).

Later Honkala, Litsyn, and Tietäväinen [8] used the so-called Elias argument for further improvements.

Very recently it was shown by Honkala, Laihonen, and Litsyn [7] that by using discrete Chebyshev polynomials instead of regular ones we get even better bounds. The bound given was as follows.

THEOREM 1.1 (See [7]). *Let $\mathcal{C} = (C_n)_{n=1}^{\infty}$ be a sequence of binary linear codes of length n, dual distance $d' = d'(n)$, and covering radius $R = R(n)$, where $R/n \to \rho$ and $d'/n \to \delta' \in (0, 1/2)$ when $n \to \infty$. Then ρ is upperbounded by any α satisfying the inequality*

$$H_2\left(\frac{1}{2} - \sqrt{\delta'(1-\delta')}\right) < \alpha H_2\left(\frac{\xi}{\alpha}\right) + (\delta' + \xi)H_2\left(\frac{\xi}{\delta' + \xi}\right) + (1-\zeta)H_2\left(\frac{\alpha - \xi}{1-\zeta}\right) - \frac{1}{2}(\alpha + 1 - \zeta - \delta')H_2\left(\frac{2\alpha}{\alpha + 1 - \zeta - \delta'}\right) - \alpha,$$

where $\zeta = \frac{1}{2} - \sqrt{\frac{1}{2}(\frac{1}{2} - \delta')}$ *and*

$$\xi = \frac{\alpha}{2(\alpha + 1 - \zeta - \delta')}(\alpha - 2\delta' + \sqrt{4\delta'^2 + \alpha^2 + 4\delta'(1-\zeta-\delta')}).$$

The aim of this paper is to improve on the bounds mentioned above. It is done by combining a suitable discrete Chebyshev polynomial with information on the least zero of a Hahn polynomial and with a bound on the cardinality of a constant weight code. This helps us to push the point of "meaningless weights" of the dual code further from the Elias range, thus giving better bounds. Moreover, for self-complementary codes we can utilize the fact that the dual code is an even-weight code (in fact, our method can be applied as well to the situation where all the weights in dual code are divisible by some other integer). This enables us to improve on Tietäväinen's bound for self-complementary codes (see [19, Theorem 3]—observe that it is asymptotically the same as (1.1) above).

2. New bounds for binary linear codes. Let $C \subset \mathbb{F}_2^n$ be a linear code with covering radius R and dual distance d' ($d' \leq \lfloor n/2 \rfloor$). We define the weight distribution of the translate $\mathbf{x} + C$ ($\mathbf{x} \in \mathbb{F}_2^n$) to be the $(n+1)$-tuple $\mathbf{A}(\mathbf{x}) = (A_0(\mathbf{x}), \ldots, A_n(\mathbf{x}))$ where $A_i(\mathbf{x}) = |\{\mathbf{c} \in C | \ w(\mathbf{x} + \mathbf{c}) = i\}|$. The MacWilliams transform [1, p. 227] of $\mathbf{A}(\mathbf{x})$ is denoted by $\mathbf{A}'(\mathbf{x}) = (A_0'(\mathbf{x}), \ldots, A_n'(\mathbf{x}))$.

It is known that we can approximate the covering radius of a code according to the following theorem (see [1, Chapter 8]). Let $K_t(x)$ be the Krawtchouk polynomial [1, p. 25] of degree t.

THEOREM 2.1 (See [7, 1]). *Let $f(x) = \sum_{t=0}^{n} \alpha_t K_t(x)$, where $\alpha_t \leq 0$ for all $t = r+1, \ldots, n$. Assume that for each $\mathbf{b} \in \mathbb{F}_2^n$*

$$f(0) + \sum_{i=1}^{n} A_i'(\mathbf{b}) f(i) > 0.$$

Then the covering radius R of the code C is at most r.

In our situation it is convenient to use the corollary below, which is a consequence of the fact that $|A_i'(\mathbf{b})| \leq A_i'$ [1, Chapter 8]. Here A_i' denotes the number of words of weight i in the dual code $C^{\perp}$. Let l and e be positive integers which satisfy $l + e < d'$.

(Notice that it would be enough for the corollary to demand that $l+e \le n-d'$, but for later use we shall prefer this stronger condition.)

COROLLARY 2.2. *Let $f(x)$ be a polynomial of degree r with real coefficients. Assume that $l+e < d'$ and*

$$f(0) - \sum_{i=d'}^{n-l-e} A_i' |f(i)| - \sum_{i=n-l-e+1}^{n-l-1} A_i' |f(i)| - \sum_{i=n-l}^{n} A_i' |f(i)| > 0. \tag{2.1}$$

Then $R \le r$.

In order to use the previous corollary efficiently we would like to find a polynomial of low degree such that $|f(i)|$ is small compared to $f(0)$ whenever $i \neq 0$. We define

$$D_r^m(x) = \sum_{j=0}^{r} (-1)^j \binom{r}{j} \binom{x}{j} \binom{m-x}{r-j}$$

to be the discrete Chebyshev polynomial of degree r on the interval $[0, m]$ (see [17, p. 33] or [14, p. 267]). It is known that these polynomials are orthogonal in the following sense [17, p. 34] (see also [14, p. 268]):

$$\sum_{i=0}^{m} D_r^m(i) D_l^m(i) = \delta_{r,l} \binom{m+r+1}{2r+1} \binom{2r}{r}, \tag{2.2}$$

where $\delta_{r,l}$ is the Kronecker symbol. Because $\max_{i \in \{0,\ldots,m\}} x_i^2 \le \sum_{i=0}^{m} x_i^2$, one gets by choosing $r = l$ in (2.2) that

$$\max_{i \in \{0,\ldots,m\}} |D_r^m(i)| \le \sqrt{\binom{m+r+1}{2r+1} \binom{2r}{r}}. \tag{2.3}$$

The discrete Chebyshev polynomial [17, p. 39] gives the minimum value of $\sum_{i=0}^{m} p(i)^2$ among the polynomials $p(x)$ of degree r with leading coefficient 1.

Let $\mathcal{C} = (C_n)_{n=1}^{\infty}$ be a sequence of binary linear codes of length n, dimension $k_{\dim} = k_{\dim}(n)$, dual distance $d' = d'(n)$, and covering radius $R = R(n)$, where $R/n \to \rho$ and $d'/n \to \delta'$ when $n \to \infty$. Assume also that $0 < \delta' < 1/2$.

Let λ be a positive number less than the Elias range, i.e., $0 < \lambda < \zeta$, and we choose $l = \lfloor \lambda n \rfloor$ and $e = \lfloor \epsilon n \rfloor$, where $\epsilon > 0$. Assume that $\lambda + \epsilon < \delta'$. In (2.1) we take the polynomial

$$\mathcal{F}(x) := \frac{D_r^m(x-d')}{D_r^m(-d')},$$

where $m = n - l - d' - e$. Observe that

$$D_r^m(-d') = \sum_{j=0}^{r} \binom{r}{j} \binom{d'+j-1}{j} \binom{m+d'}{r-j}. \tag{2.4}$$

Our goal in the next two lemmas is to show that the last two sums on the left-hand side of (2.1) tend to zero as n approaches to infinity. Denote $r = \lceil \gamma n \rceil$. Let us first check the last sum.

LEMMA 2.3. *We have*

$$\sum_{i=n-l}^{n} A_i' |\mathcal{F}(i)| \to 0, \quad n \to \infty.$$

Proof. Evidently,

$$\sum_{i=n-l}^{n} A_i^{'}|\mathcal{F}(i)| \leq \max_{i=n-l,\ldots,n} |\mathcal{F}(i)| \sum_{i=n-l}^{n} A_i^{'}.$$

It is known (see, e.g., [11, p. 61]) that

$$\sum_{i\geq n-l} A_i^{'} \stackrel{<}{\sim} \frac{\frac{1}{2}\delta'}{\lambda^2-\lambda+\frac{1}{2}\delta'}, \qquad n\to\infty. \tag{2.5}$$

The notation $F(n) \stackrel{<}{\sim} G(n)$ means that $F(n) \leq G(n)(1+\varepsilon(n))$, where $\varepsilon(n) \to 0$ as $n \to \infty$.

Let $0 \leq j \leq r \leq m$. By the symmetry of discrete Chebyshev polynomials with respect to $m/2$ we have $|D_r^m(n-d')| = |D_r^m(-l-e)|$. For the ratio of corresponding summands of $D_r^m(-l-e)$ and $D_r^m(-d')$ we get according to (2.4) that

$$\frac{\binom{r}{j}\binom{l+e+j-1}{j}\binom{m+l+e}{r-j}}{\binom{r}{j}\binom{d'+j-1}{j}\binom{m+d'}{r-j}} \leq \left(\frac{l+e+j}{d'+j}\right)^j \left(\frac{l+e+m}{d'+m}\right)^{r-j} \leq \left(\frac{m+l+e}{m+d'}\right)^r.$$

Therefore, since $\lambda+\epsilon<\delta'$, one has $D_r^m(-l-e)/D_r^m(-d') \to 0$ as $n\to\infty$. Combining this with (2.5) finishes the proof of the claim. ∎

By (2.5) we know that the cardinality of a code (with the minimum distance d') inside a Hamming ball of radius less than Elias range is subexponential in n. In order to deal with the weights, which are more than Elias range away from the "all-one" word, we need to develop a different approach.

Let us now consider the sum

$$\sum_{i=n-l-e+1}^{n-l-1} A_i^{'}|\mathcal{F}(i)|.$$

We denote the maximum cardinality of constant weight code of length n, minimum distance d, and weight w by $M(n,d,w)$. Consequently,

$$\sum_{i=n-l-e+1}^{n-l-1} A_i^{'}|\mathcal{F}(i)| \leq \max_{i=n-l-e+1,\ldots,n-l-1} |\mathcal{F}(i)| \sum_{i=n-l-e+1}^{n-l-1} M(n,d',i).$$

Since the method of the preceding corollary works best for relatively large values of (normalized) dual distance, we upperbound the value $M(n,d',w)$ by the inequality stated below (see [13, inequality (4.11)]):

$$M(n,d,w) \leq \binom{n}{k} \frac{(n^2-(2k-1)n-2k)^2(w-k)(n-w-k)}{x_1^{(k+1)}(k+1)(n-k+1)(n-2k-1)(n-2k)(n-2k+1)} \tag{2.6}$$

provided that $k < w \leq n/2$ and $x_1^{(k)} \leq d/2$, where $x_1^{(k)}$ is the least zero of the Hahn polynomial

$$J_k(x) = \sum_{j=0}^{k} (-1)^j \frac{\binom{k}{j}\binom{n+1-k}{j}}{\binom{w}{j}\binom{n-w}{j}} \binom{x}{j}.$$

The Hahn polynomials form an orthogonal family of polynomials [10]. In general, the exact value of $x_1^{(k)}$ is not known and the usual asymptotic approximations (cf. [13]) are not strong enough for our purposes. However, Levenshtein [10, subsection 5.2] has recently given in his chapter of *Handbook of Coding Theory* general estimates on extreme roots of orthogonal polynomials. We use his estimate for Hahn polynomials. With some effort one can deduce that (see [10, Lemma 5.13])

$$x_1^{(k)} \le \frac{1}{2} w \left(\frac{2}{l'} + \frac{2(n+2)^3}{l'k(n-2k)^2} \right) + \left(1 - \frac{2}{l'}\right) \left(\frac{w(n-w) - (k - \lfloor \sqrt{k} \rfloor)(n - (k - \lfloor \sqrt{k} \rfloor))}{n + 2\sqrt{(k - \lfloor \sqrt{k} \rfloor)(n - (k - \lfloor \sqrt{k} \rfloor))}} \right) \tag{2.7}$$

provided that $k > 1$ and $k < \frac{1}{2}(n + 1 - \sqrt{(n+1)^2 - 4w(n-w)})$. Here $l' = \lfloor \sqrt{k} \rfloor + 1$. We denote the right-hand side of (2.7) by $U(n, w, k)$. Now we are in a position to investigate the second sum of (2.1).

Lemma 2.4. *Suppose that*

$$\frac{\alpha(1-\alpha) - \beta(1-\beta)}{1 + 2\sqrt{\beta(1-\beta)}} < \frac{\delta'}{2}, \tag{2.8}$$

where $\alpha = \lambda + \epsilon$ *and* $0 < \beta < \lambda < 1/2$. *Then*

$$\max_{i=n-l-e+1,\ldots,n-l-1} |\mathcal{F}(i)| \sum_{i=n-l-e+1}^{n-l-1} M(n, d', i) \to 0, \qquad n \to \infty,$$

if

$$\left(\frac{1 - \lambda - \delta'}{1 - \lambda - \epsilon} \right)^{\gamma} 2^{H_2(\beta)} < 1. \tag{2.9}$$

Proof. Let $k = \lceil \beta n \rceil$ where β satisfies (2.8). Denote $w' = l + e - 1$. Since

$$\frac{U(n, w', k)}{n} \to \frac{\alpha(1-\alpha) - \beta(1-\beta)}{1 + 2\sqrt{\beta(1-\beta)}}, \quad n \to \infty,$$

we know by virtue of (2.8) that $U(n, w', k) < d'/2$ for large enough n. Furthermore, since $U(n, w, k)$ is increasing with respect to w (n and k are fixed), the same $k = \lceil \beta n \rceil$ satisfies $U(n, w, k) < d'/2$ for smaller values of w ($l + 1 \le w \le w'$). It should be noticed that the assumption $\beta < \lambda$ guarantees that the condition $k < \frac{1}{2}(n + 1 - \sqrt{(n+1)^2 - 4w(n-w)})$ holds also for small w's when n is sufficiently large. Therefore, one gets according to (2.7) and (2.6) that for large enough n

$$M(n, d', w) \le (x_1^{(k+1)})^{-1} \binom{n}{k} n^6 \tag{2.10}$$

for *all* weights $l + 1 \le w \le w'$. It follows from $\beta < \lambda$ also that $k + 1 < \frac{1}{2}(n + 1 - \sqrt{(n+1)^2 - 4w(n-w)})$ and so it implies that $x_1^{(k+1)} \ge 1$. Consequently, since $M(n, d, w) = M(n, d, n - w)$ and in addition

$$\max_{i=n-l-e+1,\ldots,n-l-1} |\mathcal{F}(i)| \le \frac{D_r^m(-e+1)}{D_r^m(-d')} \le \left(\frac{m+e}{m+d'} \right)^r,$$

we obtain (recalling from [1, p. 33] that $\binom{n}{k} \le 2^{nH_2(k/n)}$)

$$\begin{aligned}
&\max_{i=n-l-e+1,\dots,n-l-1} |\mathcal{F}(i)| \sum_{i=n-l-e+1}^{n-l-1} M(n,d',i) \\
&\le \left(\frac{m+e}{m+d'}\right)^r (e-1)\binom{n}{k} n^6 \\
&\le \left(\left(\frac{n-l-d'}{n-l-e}\right)^\gamma 2^{H_2(k/n)+\frac{1}{n}\log_2 n^6(e-1)}\right)^n .
\end{aligned}$$

Therefore, if $(\frac{1-\lambda-\delta'}{1-\lambda-\epsilon})^\gamma 2^{H_2(\beta)} < 1$, the assertion is true. □

Suppose now that β and γ are chosen so that they satisfy the assumptions in the previous lemma. Denote

$$\sigma_n = -\sum_{i=n-l-e+1}^{n-l-1} A_i'|\mathcal{F}(i)| - \sum_{i=n-l}^{n} A_i'|\mathcal{F}(i)|.$$

Combining lemmas with (2.1) we conclude that $R \le r$ if

$$1 - \max_{i=d',\dots,n-l-e} |\mathcal{F}(i)| 2^{n-k_{\rm dim}} + \sigma_n > 0,$$

where $\lim_{n\to\infty} \sigma_n = 0$. By virtue of (2.3)

$$1 - \max_{i=d',\dots,n-l-e} |\mathcal{F}(i)| 2^{n-k_{\rm dim}} \ge 1 - \frac{\sqrt{\binom{m+r+1}{2r+1}\binom{2r}{r}}}{D_r^m(-d')} 2^{n-k_{\rm dim}}.$$

Thus, $R \le r$ for large enough n if

$$2^{n-k_{\rm dim}} \le \frac{1}{2}\frac{D_r^m(-d')}{\sqrt{\binom{m+r+1}{2r+1}\binom{2r}{r}}}.$$

Evidently, any summand of the sum (2.4) provides a lower bound on $D_r^m(-d')$. With some effort one can show that asymptotically the best choice is the summand corresponding to the index

$$j = \left\lfloor \frac{r}{2(r+m)}(r - 2d' + \sqrt{4d'^2 + r^2 + 4d'm}) \right\rfloor .$$

Consequently, $R \le r$ for sufficiently large n if

$$2^{n-k_{\rm dim}+1} \le \binom{r}{j}\binom{d'+j-1}{j}\binom{m+d'}{r-j}\binom{m+r+1}{2r+1}^{-1/2}\binom{2r}{r}^{-1/2}. \tag{2.11}$$

Assume that $\lim_{n\to\infty} j/n = \xi$. It is well known (see [11, p. 21]) that

$$\lim_{a\to\infty} \frac{1}{a}\log_2\binom{a}{b} = H_2(\tau),$$

where $\lim_{n\to\infty} b/a = \tau$. Because [13]

$$\frac{n-k_{\rm dim}}{n} \lesssim H_2\left(1/2 - \sqrt{\delta'(1-\delta')}\right),$$

TABLE 2.1
A comparison of bounds for binary linear codes.

	(1.1)	[8]	Theorem 1.1	Theorem 2.5
$\delta' = 0.28$	0.1530	0.1520	0.1504	0.1442

we obtain that $\rho \le \gamma$ if

$$\begin{aligned} 0 < \gamma H_2\left(\frac{\xi}{\gamma}\right) &+ (\delta'+\xi)H_2\left(\frac{\xi}{\delta'+\xi}\right) + (1-\lambda-\epsilon)H_2\left(\frac{\gamma-\xi}{1-\lambda-\epsilon}\right) \\ &-\frac{1}{2}(1-\lambda-\delta'-\epsilon+\gamma)H_2\left(\frac{2\gamma}{1-\lambda-\delta'-\epsilon+\gamma}\right) \\ &-\gamma - H_2(1/2-\sqrt{\delta'(1-\delta')}) \end{aligned} \tag{2.12}$$

provided that the assumptions of the preceding lemma hold. Recalling that this is true for all $\lambda < \zeta$ and the left-hand side of (2.9) is continuous with respect to λ, it is not hard to check that we get the following main theorem.

THEOREM 2.5. *Let $\mathcal{C} = (C_n)_{n=1}^{\infty}$ be a sequence of binary linear codes of length n, dual distance $d' = d'(n)$, and covering radius $R = R(n)$, where $R/n \to \rho$ and $d'/n \to \delta'$ when $n \to \infty$. Assume that $0 < \delta' < 1/2$ and $\alpha = \zeta + \epsilon$. Suppose that*

$$\frac{\alpha(1-\alpha)-\beta(1-\beta)}{1+2\sqrt{\beta(1-\beta)}} < \frac{\delta'}{2}$$

and $0 < \beta < \zeta$. Assume also that $\zeta + \epsilon < \delta'$ and

$$\left(\frac{1-\zeta-\delta'}{1-\zeta-\epsilon}\right)^{\gamma} 2^{H_2(\beta)} < 1.$$

Then $\rho \le \gamma$ for every γ satisfying inequality (2.12) *when λ is replaced by ζ.*

This result is better than Tietäväinen's bound when $\delta' \ge 0.271$, and it improves on Theorem 1.1 (see Table 2.1).

3. Improvement for self-complementary codes. Let us now consider a sequence $\mathcal{C} = (C_n)_{n=1}^{\infty}$ where all the codes are linear and self-complementary. As before, suppose $\lim_{n\to\infty} R/n = \rho$ and $\lim_{n\to\infty} d'/n = \delta'$. Denote the cardinality of C_n by $k_{\dim} = k_{\dim}(n)$. Now we may combine the previous approach with the fact that the dual codes are of even weight. Moreover, this method applies to the situation where all the weights in dual codes are divisible by some other integer as well. The goal of this section is to find a better upper bound than the one for self-complementary codes in [19, Theorem 3].

Denote $m = n-l-e-d'$, where $l = \lfloor \lambda n \rfloor$, $e = \lfloor \epsilon n \rfloor$. Suppose $\lambda < \zeta$ and $\lambda + \epsilon < \delta'$. We now apply the modified polynomial

$$\mathcal{F}(x) := \frac{D_r^{\lceil \frac{m}{2} \rceil}((x-d')/2)}{D_r^{\lceil \frac{m}{2} \rceil}(-d'/2)}$$

to (2.1). Since

$$\frac{D_r^{\lceil \frac{m}{2} \rceil}(-\lceil \frac{l+e}{2} \rceil)}{D_r^{\lceil \frac{m}{2} \rceil}(-d'/2)} \le \left(\frac{\lceil \frac{l+e}{2} \rceil + \lceil \frac{m}{2} \rceil}{d'/2 + \lceil \frac{m}{2} \rceil}\right)^r,$$

we may deduce that Lemma 2.3 holds also for this polynomial.

Similarly, because

$$\frac{D_r^{\lceil \frac{m}{2} \rceil}(-\lceil \frac{e}{2} \rceil)}{D_r^{\lceil \frac{m}{2} \rceil}(-d'/2)} \leq \left(\frac{\lceil \frac{e}{2} \rceil + \lceil \frac{m}{2} \rceil}{d'/2 + \lceil \frac{m}{2} \rceil} \right)^r,$$

Lemma 2.4 is valid for our current polynomial.

Moreover,

$$\begin{aligned} &1 - \sum_{i=d'}^{n-l-e} A_i^{'} |\mathcal{F}(i)| \\ &\geq \begin{cases} 1 - \max_{i=d',d'+2,\dots,n-l-e} |\mathcal{F}(i)| 2^{n-k_{\dim}} & \text{if } 2 \mid n-l-e, \\ 1 - \max_{i=d',d'+2,\dots,n-l-e+1} |\mathcal{F}(i)| 2^{n-k_{\dim}} & \text{otherwise}, \end{cases} \\ &= 1 - \frac{2^{n-k_{\dim}}}{D_r^{\lceil \frac{m}{2} \rceil}(-d'/2)} \max_{i=0,1,\dots,\lceil \frac{m}{2} \rceil} |D_r^{\lceil \frac{m}{2} \rceil}(i)|. \end{aligned}$$

Now the same arguments as in Theorem 2.5 yield the statement below.

THEOREM 3.1. *Let* $\mathcal{C} = (C_n)_{n=1}^{\infty}$ *be a sequence of linear self-complementary codes of length* n*, dual distance* $d' = d'(n)$*, and covering radius* $R = R(n)$*, where* $R/n \to \rho$ *and* $d'/n \to \delta'$ *when* $n \to \infty$*. Assume that* $0 < \delta' < 1/2$ *and* $\alpha = \zeta + \epsilon$*. Suppose that*

$$\frac{\alpha(1-\alpha) - \beta(1-\beta)}{1 + 2\sqrt{\beta(1-\beta)}} < \frac{\delta'}{2}$$

and $0 < \beta < \zeta$*. Assume also that* $\zeta + \epsilon < \delta'$ *and* $(\frac{1-\zeta-\delta'}{1-\zeta-\epsilon})^{\gamma} 2^{H_2(\beta)} < 1$*. Then any* γ *satisfying*

$$\begin{aligned} 0 < \gamma H_2\left(\frac{\xi}{\gamma}\right) &+ (\delta'/2 + \xi) H_2\left(\frac{\xi}{\delta'/2 + \xi}\right) + \frac{1}{2}(1 - \zeta - \epsilon) H_2\left(\frac{2(\gamma - \xi)}{1 - \zeta - \epsilon}\right) \\ &- \frac{1}{2}\left(\frac{1}{2}(1 - \zeta - \delta' - \epsilon) + \gamma\right) H_2\left(\frac{2\gamma}{\frac{1}{2}(1 - \zeta - \delta' - \epsilon) + \gamma}\right) \\ &- \gamma - H_2(1/2 - \sqrt{\delta'(1 - \delta')}), \end{aligned} \tag{3.1}$$

where

$$\xi = \frac{\gamma}{2(\gamma + \frac{1}{2}(1 - \zeta - \delta' - \epsilon))} (\gamma - \delta' + \sqrt{\delta'^2 + \gamma^2 + \delta'(1 - \zeta - \delta' - \epsilon)})$$

gives an upper bound on ρ.

This bound improves on Tietäväinen's result when $\delta' \geq 0.265$. It is also better than Theorem 1.1 (at $\delta' = 0.28$ the improvement is 0.01).

Acknowledgments. The first author is very grateful to professor Levenshtein for sending the chapter of *Handbook of Coding Theory*. We would also like to thank I. Honkala and A. Tietäväinen for fruitful discussions.

REFERENCES

[1] G. COHEN, I. HONKALA, S. LITSYN, AND A. LOBSTEIN, *Covering Codes*, Elsevier, Amsterdam, 1997.

[2] C. DELORME AND P. SOLÉ, *Diameter, covering index, covering radius and eigenvalues*, European J. Combin., 12 (1991), pp. 95–108.

[3] P. Delsarte, *Four fundamental parameters of a code and their combinatorial significance,* Inform. and Control, 23 (1973), pp. 407–438.

[4] G. Fazekas and V. I. Levenshtein, *On upper bounds for code distance and covering radius of designs in polynomial metric spaces*, J. Combin. Theory Ser. A, 70 (1995), pp. 267–288.

[5] T. Helleseth, *On the covering radius of cyclic linear codes and arithmetic codes*, Discrete Appl. Math., 11 (1985), pp. 157–173.

[6] T. Helleseth, T. Kløve, and J. Mykkeltveit, *On the covering radius of binary codes,* IEEE Trans. Inform. Theory, 24 (1978), pp. 627–628.

[7] I. Honkala, T. Laihonen, and S. Litsyn, *On covering radius and discrete Chebyshev polynomials*, Appl. Algebra Engrg. Comm. Comput., 8 (1997), pp. 395–402.

[8] I. Honkala, S. Litsyn, and A. Tietäväinen, *On algebraic methods in covering radius problems*, in Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, Lecture Notes in Comput. Sci. 948, G. Cohen, M. Giusti, and T. Mora, eds., Springer-Verlag, Berlin, Heidelberg, New York, 1995, pp. 21–32.

[9] T. Laihonen and S. Litsyn, *On upper bounds for minimum distance and covering radius of non-binary codes*, Des. Codes Cryptogr., 14 (1998), pp. 71–80.

[10] V. I. Levenshtein, *Universal bounds for codes and designs*, in Handbook of Coding Theory, V. S. Pless, W. C. Huffman, and R. A. Brualdi, eds., to appear.

[11] J. van Lint, *Introduction to Coding Theory,* Springer-Verlag, New York, Berlin, 1982.

[12] S. Litsyn and A. Tietäväinen, *Upper bounds on the covering radius of a code with a given dual distance,* European J. Combin., 173 (1996), pp. 265–270.

[13] R. J. McEliece, E. R. Rodemich, H. C. Rumsey, and L. R. Welch, *New upper bounds on the rate of the code via the Delsarte-MacWilliams inequalities*, IEEE Trans. Inform. Theory, 23 (1977), pp. 157–166.

[14] W. Milne, *Numerical Calculus,* Princeton University Press, Princeton, NJ, 1949.

[15] P. Solé, *Asymptotic bounds on the covering radius of binary codes,* IEEE Trans. Inform. Theory, 36 (1990), pp. 1470–1472.

[16] P. Solé and P. Stokes, *Covering radius, codimension, and dual-distance width*, IEEE Trans. Inform. Theory, 39 (1993), pp. 1195–1203.

[17] G. Szegö, *Orthogonal Polynomials,* Amer. Math. Soc. Colloq. Publ. 23, revised ed., AMS, Providence, RI, New York, 1959.

[18] A. Tietäväinen, *Codes and character sums*, in Coding Theory and Applications, Lecture Notes in Comput. Sci. 388, Springer-Verlag, New York, 1989, pp. 3–12.

[19] A. Tietäväinen, *An upper bound on the covering radius as a function of its dual distance*, IEEE Trans. Inform. Theory, 36 (1990), pp. 1472–1474.

[20] A. Tietäväinen, *Covering radius and dual distance*, Des. Codes Cryptogr., 1 (1991), pp. 31–46.

SIAM J. DISCRETE MATH.
Vol. 12, No. 2, pp. 252–261

INTERCONNECTING HIGHWAYS*

DING-ZHU DU†, FRANK K. HWANG‡, AND GUOLIANG XUE§

Abstract. We present the problem of constructing roads of minimum total length to interconnect n highways under the constraint that the roads can intersect each highway only at one point in a designated interval which is a line segment. We present a set of optimality conditions for the problem and show how to construct a solution to meet this set of optimality conditions.

Key words. interconnecting networks, optimality conditions, Steiner trees

AMS subject classifications. 05C05, 90C27

PII. S089548019732653X

1. Introduction. We present the problem of constructing roads of minimum total length to interconnect n existing highways H_1, H_2, ..., H_n under the constraint that the roads can intersect H_i only at one point, called an *exit* of H_i, in a designated interval I_i. To avoid unnecessary complexity, we assume that all I_i's are disjoint. In this paper, we consider the case where I_i is a line segment, including the two extreme cases where I_i is a point or a line. The case where I_i is a point for all $i = 1, 2, \ldots, n$ is the Steiner minimum tree problem which is NP-hard [5]. Thus, the current problem is also NP-hard. Some special cases for $n = 3$ have been studied by Chen [3] and Weng [15]. More applications and the relation to facility allocation problems can be found in [3, 8, 18].

We will first establish a set of optimality conditions and then show how to construct a solution to meet this set of conditions by generalizing Melzak's construction for Steiner trees. Finally, we will use those results to determine global optimal solutions for $n = 2$ and $n = 3$.

2. Optimality conditions. Let us call each intersection of roads, which is not an exit, a *Steiner point*. Consider an optimal solution for the problem of interconnecting highways. Clearly, this solution must be the Steiner minimum tree for the n exits at the current positions. Thus, it must have properties for Steiner minimum trees as stated in the following [6, 8].

LEMMA 2.1. *An optimal solution for the problem of interconnecting highways must satisfy the following conditions:*

(a) *Every Steiner point has degree three (Figure* 2.1(a)*).*

(b) *Two roads meeting at a point form an angle of at least* $120°$ *(Figure* 2.1(b)*).*

Since each exit can move in the designated interval I_i, we have additional optimality conditions at exits.

*Received by the editors July 18, 1997; accepted for publication May 7, 1998; published electronically April 29, 1999.

http://www.siam.org/journals/sidma/12-2/32653.html

†Department of Computer Science, University of Minnesota, Minneapolis, MN 55455, and Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing, China (dzd@cs.umn.edu). The research of this author was supported in part by NSF grants CCR-9530306 and OSR-9350540.

‡Department of Applied Mathematics, Chiao-Tung University, Hsin-Chu 30050, Taiwan, Republic of China (fhwang@math.nctu.edu.tw).

§Department of Computer Science, The University of Vermont, Burlington, VT 05405-0156 (xue@ecs.uvm.edu). The research of this author was supported in part by US Army Research Office grant DAAH04-96-1-0233 and by NSF grants ASC-9409285 and OSR-9350540.

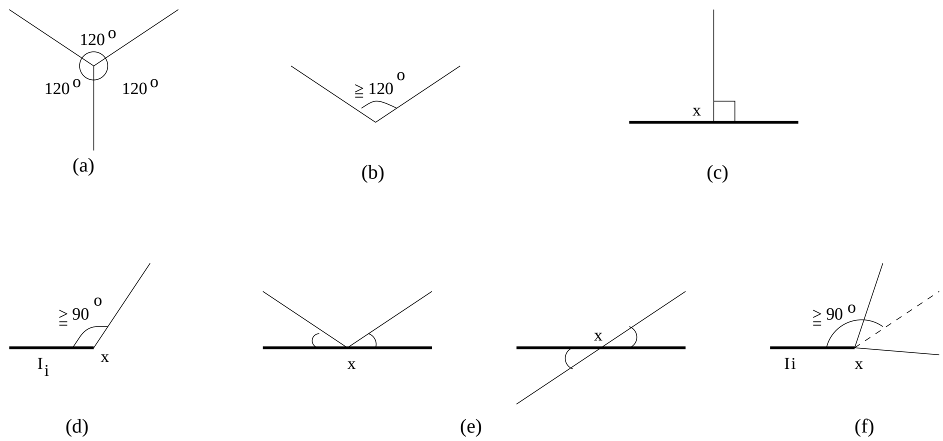


FIG. 2.1. *Optimality conditions.*

LEMMA 2.2. *Let x be an exit in interval I_i. An optimal solution for the problem of interconnecting highways must satisfy the following conditions:*

(c) *If exit x is connected to only one road and x is an interior point of I_i, then this road is perpendicular to I_i (Figure* 2.1(c)*).*

(d) *If exit x is connected to only one road and x is an endpoint of I_i, then this road together with I_i forms an angle of at least* $90°$ *(Figure* 2.1(d)*).*

(e) *If exit x is connected to exactly two roads and x is an interior point of I_i, then the angle formed by one road and a part of I_i equals the angle formed by the other road and the other part of I_i (Figure* 2.1(e)*).*

(f) *If exit x is connected to exactly two roads and x is an endpoint of I_i, then the bisector of the relative angle of the two roads and I_i form an angle of at least* $90°$ *(Figure* 2.1(f)*).*

Proof. Suppose $I_i = [A, B]$.

(c) Suppose exit x is connected to only one road (x, C). Since x is an interior point of $[A, B]$, if xC is not perpendicular to $[A, B]$, then either $\angle AxC < 90°$ or $\angle BxC < 90°$. In the former case, moving x in direction xA would decrease distance xC (Figure 2.2). In the latter case, moving x in direction xB would decrease xC. Thus, x is not at an optimal position.

(d) A similar argument to (c) could apply (Figure 2.2).

(e) Suppose that exit x is connected to two roads (x, C) and (x, D). If C and D lie on different sides of line AB, then C, x, and D must be on the same line. Otherwise, a little perturbation would make a shorter solution (Figure 2.3). If C and D are on the same side of line AB, then find the point C' which is symmetric to C with respect to line AB. Then, C', x, and D must be on the same line. Otherwise, a little perturbation would make a shorter solution (Figure 2.3). In both case, we have $\angle AxC = \angle BxD$.

(f) An argument similar to (e) could apply. ∎

We call a tree satisfying conditions (a)–(f) a *legitimate tree*. A legitimate tree is *full* if every exit is a leaf.

THEOREM 2.3. *The optimal solution of the highway interconnection problem must be a legitimate tree.*

Since a legitimate tree is a Steiner tree for a current position of exits, we may

FIG. 2.2. *Moving x in direction xA decreases xC.*

FIG. 2.3. *A little perturbation would shorten the tree.*

extend some concepts about Steiner trees to legitimate trees as follows: A *topology* is the graph structure of a legitimate tree. The topology is full if the legitimate tree is full. A topology can be degenerated by shrinking an edge between a Steiner point and an exit. A topology is called a *degenerate* one of another topology if the former can be obtained from the latter by a sequence of degenerate operations.

THEOREM 2.4. *Among a full topology and its degenerate topologies, if there exists one with which the legitimate tree exists, then it is minimum among all trees with the full topology and its degenerate ones.*

Proof. Consider the problem of finding the shortest one among all trees under a full topology (including its degenerate topologies) interconnecting n exits each in a designated interval. This problem has a convex objective function with respect to coordinates of Steiner points and exits, which is a sum of Euclidean distances, and linear constraints on coordinates of exits. Therefore, it is a convex programming. Any local optimal solution is also a global optimal solution. In the following, we will show that if a legitimate tree with the full topology or its degenerate one exists, then it is a local minimum for the convex programming. To do so, we show that at the legitimate tree, every feasible direction is not descending. That is, the directional derivative of the objective function is nonnegative.

Let t be the full topology and $E(t)$ the edge set of t. Then, the objective function of the convex programming is

$$f(V(t)) = \sum_{(u,v)\in E(t)} \|u - v\|,$$

where $V(t)$ is the vertex set of t and all coordinates of vertices are variables of function f. A feasible direction ΔV of $V(t)$ consists of moving direction Δv for every vertex $v \in V(t)$. For a Steiner point, every direction can be feasible. For an exit, only direction along the exit's interval can be feasible. Suppose the feasible direction ΔV of $V(t)$ has unit length. Then, its directional derivative is

$$\lim_{\lambda\to 0}\frac{f(V(t)+\lambda\Delta V)-f(V(t))}{\lambda} = \sum_{(u,v)\in E(t)}\lim_{\lambda\to 0}\frac{\|(u+\lambda\Delta u)-(v+\lambda\Delta v)\|-\|u-v\|}{\lambda}.$$

We will first calculate this derivative and then show that it is nonnegative.

For $u \neq v$,

$$\lim_{\lambda\to 0}\frac{\|(u+\lambda\Delta u)-(v+\lambda\Delta v)\|-\|u-v\|}{\lambda} = \frac{(\Delta u-\Delta v)^T(u-v)}{\|u-v\|} = \frac{(\Delta u)^T(u-v)+(\Delta v)^T(v-u)}{\|u-v\|}.$$

For $u = v$,

$$\lim_{\lambda\to 0}\frac{\|(u+\lambda\Delta u)-(v-\lambda\Delta v)\|-\|u-v\|}{\lambda} = \lim_{\lambda\to 0}\|\Delta u-\Delta v\| = \|\Delta u-\Delta v\|.$$

Now, suppose $V(t)$ is the vertex set of a legitimate tree. If u is a Steiner point and its three edges in the legitimate tree are all of nonzero length, then there are three terms involving Δu in the directional derivative,

$$\frac{(\Delta u)^T(u-v_1)}{\|u-v_1\|}+\frac{(\Delta u)^T(u-v_2)}{\|u-v_2\|}+\frac{(\Delta u)^T(u-v_3)}{\|u-v_3\|}=0,$$

since any two of three vectors $u-v_1$, $u-v_2$, and $u-v_3$ form an angle of $120°$.

If u is an exit and its only edge in the legitimate tree has nonzero length, then Δu is involved in only one term of the directional derivative,

$$\frac{(\Delta u)^T(u-v)}{\|u-v\|}\geq 0,$$

since Δu and $v-u$ form an angle of at least $90°$, that is, $(\Delta u)^T(v-u)\leq 0$.

Note that if degeneration occurs, it must occur around an exit v. If in the legitimate tree v is incident to two edges, this means that the edge (v,u) in t, where u is a Steiner point, has been shrunk to a point. If in the legitimate tree v is incident to three edges, then two edges (v,u) and (u,w), where u and w are Steiner points, have been shrunk to one point in the legitimate tree.

In the former case, i.e., edge (u,v), where u is a Steiner point and v is an exit in t, having length 0 in the legitimate tree, then the directional derivative has three terms involving Δu and Δv as follows:

$$\|\Delta u-\Delta v\|+\frac{(\Delta u)^T(u-v_2)}{\|u-v_2\|}+\frac{(\Delta u)^T(u-v_3)}{\|u-v_3\|}.$$

Denote

$$w = \frac{v_2 - u}{\|u - v_2\|} + \frac{v_3 - u}{\|u - v_3\|}.$$

Since $v_2 - u$ and $v_3 - u$ form an angle of at least 120°, we have

$$\|w\| \le 1.$$

In addition, since w and Δv form an angle of at least 90°, we have

$$(\Delta v)^T w \le 0.$$

Therefore,

$$\begin{aligned} &\|\Delta u - \Delta v\| - (\Delta u)^T w \\ &\quad \ge \|\Delta u - \Delta v\| + (\Delta v - \Delta u)^T w \\ &\quad \ge 0. \end{aligned}$$

Finally, we consider the case where, in the legitimate tree, v is incident to three edges, i.e., two edges (v, u) and (u, w) in t, where u and w are Steiner points, have been shrunk to one point in the legitimate tree. In this case, the directional derivative has five terms involving Δu, Δv, and Δw as follows:

$$\|\Delta v - \Delta u\| + \|\Delta u - \Delta w\| + \frac{(\Delta u)^T(u - v_1)}{\|u - v_1\|} + \frac{(\Delta w)^T(w - v_2)}{\|w - v_2\|} + \frac{(\Delta w)^T(w - v_3)}{\|w - v_3\|}.$$

Denote this summation by s. Note that in the legitimate tree, u and w are identical and any two of three vectors $u - v_1$, $w - v_2$, and $w - v_3$ form an angle of 120°. Thus,

$$\frac{(u - v_1)}{\|u - v_1\|} + \frac{(w - v_2)}{\|w - v_2\|} + \frac{(w - v_3)}{\|w - v_3\|} = 0.$$

It follows that

$$s = \|\Delta v - \Delta u\| + \|\Delta u - \Delta w\| + \frac{(\Delta u - \Delta w)^T(u - v_1)}{\|u - v_1\|} \ge 0.$$

By summarizing the above, we know that the directional derivative is nonnegative. This completes our proof. □

3. Generalized Melzak construction. In this section, we study the following question: If a legitimate tree exists, how do we construct it?

First, we show how to construct a legitimate tree with full topology if it exists. Let us start by recalling Melzak's construction [11].

Melzak's construction works for a Steiner tree with a full topology. In each step, it first finds a Steiner point adjacent to two exits (they are fixed in a Steiner tree problem). Then, it constructs an equilateral triangle with the two exits as its two vertices and replaces them by the third vertex (Figure 3.1), considered a new exit. After several steps, when only two exits exist, it connects them by a straight line segment and in reverse ordering, then finds all edges of the full Steiner tree.

Now, we also want to replace two exits (adjacent to the same Steiner point) by a new exit. However, a new situation is that each exit has a feasible region. (For each original exit, its feasible region is a line segment.) Thus, we also need to construct

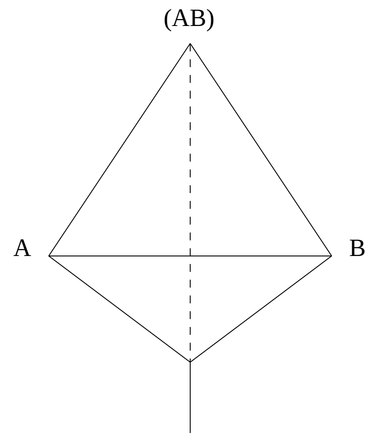


FIG. 3.1. *Melzak's construction.*

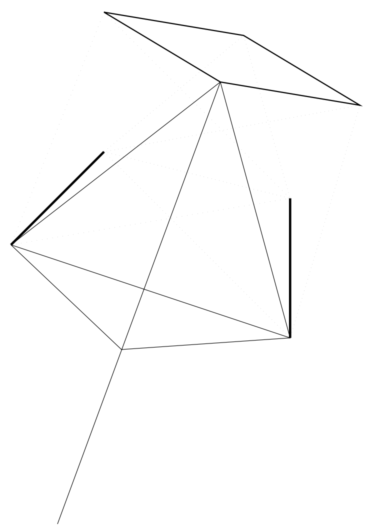


FIG. 3.2. *The feasible region of this new exit is a parallelogram.*

a feasible region for the new exit. Initially, a new exit is obtained from two original exits and the feasible region of this new exit is a parallelogram, as shown in Figure 3.2. In general, what is the feasible region for a new exit if it is obtained from k original exits through $k-1$ steps of Melzak's construction? An answer is given in the following.

Let us call a convex central symmetric $2k$-polygon a *parallel* $2k$-polygon if its $2k$ edges can be divided into k pairs of parallel edges with equal length (Figure 3.3). Note that every parallel $2k$-polygon can be covered in the following way: Choose an edge. Moving this edge along an adjacent edge will obtain a parallel 4-polygon (or a parallelogram). Moving the parallel 4-polygon along an adjacent edge will obtain a parallel 6-polygon. Continuing in this way, finally the parallel $2k$-polygon will be obtained (Figure 3.3) and all points in this parallel $2k$-polygon are covered by moving images.

THEOREM 3.1. *Let v be a new exit obtained from k original exits through $k-1$*

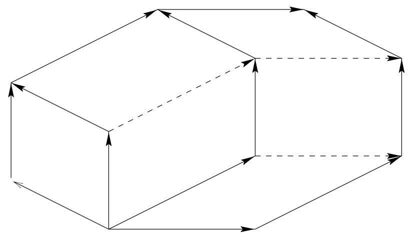


FIG. 3.3. *Parallel* $2k$*-polygon.*

steps of Melzak's construction. The feasible region of v *is a parallel* $2k$*-polygon.*

Proof. We prove it by induction on k. Suppose v is obtained from two exits u and w. Suppose u is obtained from i original exits and w is obtained from j original exits. Clearly, $i + j = k$ and $i < k$ and $j < k$. By the induction assumption, the feasible region of u is a parallel $2i$-polygon P and the feasible region of w is a parallel $2j$-polygon Q.

First, we fix u at a position in P and move w over region Q. It is easy to see that v will describe a region Q' isomorphic to Q. Actually, this region Q' can be obtained from Q by turning Q around center u in an angle of $60°$.

Next, we move u along an edge e_1 of P. As u moves, Q' will move along a certain direction and all images will cover a parallel $2(j + 1)$-polygon Q''.

Now, we move edge e_1 along an adjacent edge e_2 of P. As all images of e_1 cover a parallel 4-polygon P', all images of Q'' cover a parallel $2(j + 2)$-polygon Q'''. Continue in this way. As all points in the parallel $2i$-polygon P are covered, a parallel $2(i + j)$-polygon will be covered by images of Q' (Figure 3.4). □

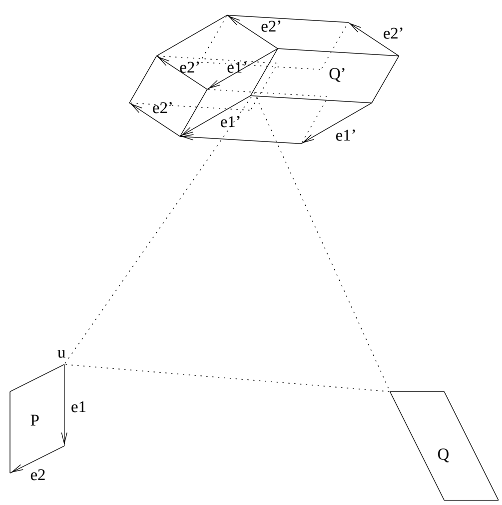


FIG. 3.4. *The proof of Theorem* 3.1.

For degenerate topology, Melzak's method for a Steiner tree is to decompose it into an edge-disjoint union of small full topologies. However, for the problem of interconnecting highways, such a decomposition does not exist since the position of an exit v connected to two roads (v, u) and (v, w) has to be determined by considering both roads. If the feasible region of u (or w) is known, then we may need to consider

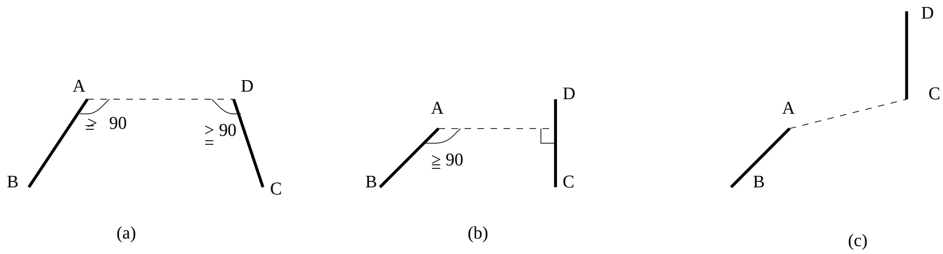


FIG. 4.1. *Two highways.*

two cases: v is in the interior of the designated interval I_i or v is at one of two endpoints of I_i. In the former case, we can replace I_i by one of its endpoints and then decompose the topology at v. In the latter case, we replace (v, u) and (v, w) by (w, u) and meanwhile replace the feasible region of u by its symmetric image with respect to I_i if u and w are in the same side of I_i.

From the above analysis, one may see that constructing the legitimate tree with a degenerate topology is much more complicated than the Steiner tree in a similar situation. It is indeed not a construction which can be finished in polynomial time with respect to the number of exits. However, Xue, Du, and Hwang [18] showed that there exists a fast way to construct a tree with length almost as short as the legitimate tree. This work draws from many previous contributions on shortest network under a given topology [7, 9, 13, 14, 16, 17].

4. Two or three highways. If our interest is only on highway interconnection, then $n = 2$ and $n = 3$ are the most practical cases. In these two cases, there is a unique full topology. Thus, a tree being legitimate is necessary and sufficient for optimality. In this section, we will apply the results from previous sections to determine the legitimate tree in these two cases.

For $n = 2$, suppose AB and CD are two line segments. Assume that BD and DA do not intersect, that is, $ABCD$ form a quadrilateral. Since $\angle A + \angle B + \angle C + \angle D = 360°$, either $\angle A + \angle D \geq 180°$ or $\angle B + \angle C \geq 180°$. Without loss of generality, assume the former occurs and $\angle A \geq \angle D$. Then $\angle A \geq 90°$. Now, we have three cases.

Case 1. $\angle D \geq 90°$. In this case, the line segment AD is the legitimate tree interconnecting AB and CD (Figure 4.1(a)).

Case 2. $\angle D < 90°$ and $\angle ACD < 90°$. In this case, draw line segment AE perpendicular to CD at E. Then AE is the legitimate tree (Figure 4.1(b)).

Case 3. $\angle D < 90°$ and $\angle ACD \geq 90°$. In this case, AC is the legitimate tree (Figure 4.1(c)).

For $n = 3$, there are four possible topologies for the legitimate tree. We first construct the one with full topology (Figure 4.2). If successful, then the work is done. If unsuccessful, then we construct the other three topologies in turn until a legitimate tree is found (Figure 4.3).

To construct the one with full topology, first replace two line segments I_1 and I_2 by a parallelogram with Melzak's construction. Then, find the shortest distance between the parallelogram and the third line segment I_3. Suppose this happens between point A in the parallelogram and point B in I_3. Note that A corresponds to two points C and D in I_1 and I_2, respectively. Draw the full Steiner tree for B, C, D. If it exists, the legitimate tree with the full topology is found; if not, then it means that the legitimate tree with the full topology does not exist.

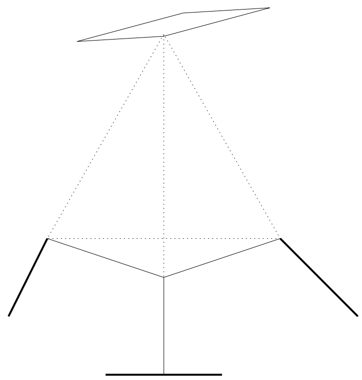


FIG. 4.2. *Three highways* (1).

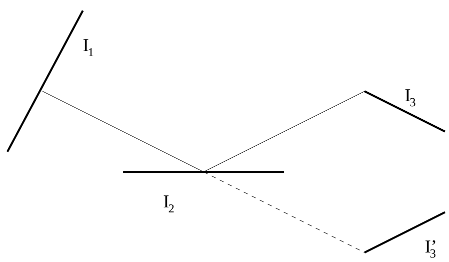


FIG. 4.3. *Three highways* (2).

To explain how to construct a legitimate tree with a degenerate topology, we may assume the topology consists of two edges between I_1 and I_2 and I_2 and I_3, respectively. Suppose I_3' is the mirror image of I_3 with respect to I_2. Find the shortest distance between I_1 and I_3 and the shortest distance between I_1 and I_3'. If a segment realizing either one of the two shortest distances intersects segment I_2, then the legitimate tree is found. If no such segment exists, then consider two endpoints of I_2. For each endpoint, find the shortest segments to connect it to I_1 and I_3, respectively. Check whether the two shortest segments form a legitimate tree. If no legitimate tree is found in this way, then it means that the legitimate tree with this degenerate topology does not exists and we should consider another degenerate topology. By Theorem 2.3, we would find a legitimate tree before all possible topologies are examined.

5. Discussion. A variation of the problem considered in this paper is to use a spanning tree instead of a Steiner tree, interconnecting points with each on a specified line segment, and to find the shortest one. So far, we do not know if such a spanning tree problem has a polynomial time solution. Therefore, approximation for the highway interconnection problem is also an open problem. No polynomial time approximation with constant performance ratio for this problem has been found, although many polynomial time approximations with good performance for Steiner minimum trees have been known [1, 2, 10].

When a new Steiner tree problem appears, one usually also considers the corresponding Steiner ratio problem. Note that a Steiner minimum tree interconnecting n line segments is also the Steiner minimum tree connecting the n exits. Let us fix these

n exits. Then, a minimum spanning tree for n line segments is not longer that the minimum spanning tree for the fixed n exits. Therefore, the ratio of lengths between the Steiner minimum tree and the minimum spanning tree for n line segments is at least $\sqrt{3}/2$. It follows that the Steiner ratio for the highway interconnection problem is the same as that for the Euclidean Steiner tree problem [4, 12].

REFERENCES

[1] S. Arora, *Polynomial time approximation schemes for Euclidean TSP and other geometric problems*, in Proc. 37th IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1996, pp. 2–11.
[2] A. Borchers and D.-Z. Du, *The k-Steiner ratio in graphs*, SIAM J. Comput., 26 (1997), pp. 857–869.
[3] G. X. Chen, *The shortest path between two points with a (linear) constraint*, Knowledge Appl. Math., 4 (1980), pp. 1–8.
[4] D.-Z. Du and F. K. Hwang, *A proof of Gilbert-Pollak's conjecture on the Steiner ratio*, Algorithmica, 7 (1992), pp. 121–135.
[5] M. R. Garey, R. L. Graham, and D. S. Johnson, *The complexity of computing Steiner minimal trees*, SIAM J. Appl. Math., 32 (1977), pp. 835–859.
[6] E. N. Gilbert and H. O. Pollak, *Steiner minimal trees*, SIAM J. Appl. Math., 16 (1968), pp. 1–29.
[7] F. K. Hwang, *A linear time algorithm for full Steiner trees*, Oper. Res. Lett., 4 (1986), pp. 235–237.
[8] F. K. Hwang, D. S. Richard, and P. Winter, *The Steiner Tree Problem*, North–Holland, Amsterdam, 1992.
[9] F. K. Hwang and J. F. Weng, *The shortest network under a given topology*, J. Algorithms, 13 (1992), pp. 468–488.
[10] M. Karpinski and A. Zelikovsky, *New approximation algorithms for the Steiner tree problems*, J. Combin. Optim., 1 (1997), pp. 47–65.
[11] Z. A. Melzak, *Companion to Concrete Mathematics*, Vol. II, John Wiley, New York, 1976.
[12] J. L. Rubinstein and J. F. Weng, *Compression theorems and Steiner ratios on spheres*, J. Combin. Optim., 1 (1997), pp. 67–78.
[13] D. Trietsch and F. K. Hwang, *An improved algorithm for Steiner trees*, SIAM J. Appl. Math., 50 (1990), pp. 244–263.
[14] A. Underwood, *A modified Melzak procedure for computing node-weighted Steiner trees*, Networks, 27 (1996), pp. 73–79.
[15] J. F. Weng, *Generalized Steiner problem and hexagonal coordinate*, Acta Math. Appl. Sinica, 8 (1985), pp. 383–397 (in Chinese).
[16] G. Xue and Y. Ye, *An efficient algorithm for minimizing a sum of Euclidean norms with applications*, SIAM J. Optim., 7 (1997), pp. 1017–1036.
[17] G. L. Xue and D.-Z. Du, *An $O(n \log n)$ average time algorithm for computing the shortest network under a given topology*, Algorithmica, to appear.
[18] G. L. Xue, D.-Z. Du, and F. K. Hwang, *Faster algorithm for shortest network under given topology*, in Topics in Semidefinite and Interior-Point Methods, Fields Inst. Commun. 18, AMS, Providence, RI, 1998, pp. 137–152.

SIAM J. Discrete Math.
Vol. 12, No. 2, pp. 262–275

BOUNDED STABLE SETS: POLYTOPES AND COLORINGS*

JEANNETTE JANSSEN† AND KYRIAKOS KILAKOS‡

Abstract. A k-stable set in a graph is a stable set of size at most k. We study the convex hull of the k-stable sets of a graph, aiming for a complete inequality description. We also consider colorings of weighted graphs by k-stable sets, aiming for a relation between the values of an optimal coloring and an optimal fractional coloring. Results for $k = 2$ and $k = 3$ as well as a number of general conjectures linking fractional and integral colorings are given.

Key words. fractional graph coloring, bounded coloring, stable set polytope

AMS subject classifications. 05C15, 90C27

PII. S089548019630978X

1. The results.

Let $G = (V, E)$ be a graph. A *stable set* of G is a set of mutually nonadjacent nodes of G. The *stable set polytope* of G is the convex hull of incidence vectors of its stable sets.

Let $\mathbf{c} \in \mathrm{Z}_+^{V(G)}$. An (integral) *coloring* of G with respect to $\mathbf{c}$, or simply a coloring of $(G, \mathbf{c})$, is an assignment ϕ of colors to the nodes of G such that

- $\phi(v)$ is a set of $c(v)$ colors, for all $v \in V(G)$, and
- $\phi(v) \cap \phi(u) = \emptyset$ for any two adjacent nodes v and u of G.

Let $\mathcal{J}$ be the family of stable sets of G. In terms of vectors, a coloring of $(G, \mathbf{c})$ is any element of

$$P_I(G, \mathbf{c}) = \left\{ \mathbf{y} \in \mathrm{Z}_+^{\mathcal{J}} : \sum_{v \in S \in \mathcal{J}} y_S = c(v), v \in V(G) \right\}.$$

A *fractional coloring* of $(G, \mathbf{c})$ is any element of

$$P(G, \mathbf{c}) = \left\{ \mathbf{y} \in \mathrm{Q}_+^{\mathcal{J}} : \sum_{v \in S \in \mathcal{J}} y_S = c(v), v \in V(G) \right\}.$$

For any given coloring $\mathbf{y}$, fractional or integral, those stable sets S for which $y_S > 0$ are its *color classes*. The interest is in finding colorings of $(G, \mathbf{c})$ that use as few colors as possible. The number of colors used by any optimal (integral) coloring of $(G, \mathbf{c})$, known as the *chromatic number* of $(G, \mathbf{c})$, is denoted by $\chi(G, \mathbf{c})$. In other words,

$$\chi(G, \mathbf{c}) = \min\{\mathbf{1} \cdot \mathbf{y} : \mathbf{y} \in P_I(G, \mathbf{c})\},$$

where $\mathbf{1}$ is the all-one row vector. The *fractional chromatic number* of $(G, \mathbf{c})$, denoted $\eta(G, \mathbf{c})$, is the number

$$\eta(G, \mathbf{c}) = \min\{\mathbf{1} \cdot \mathbf{y} : \mathbf{y} \in P(G, \mathbf{c})\}.$$

*Received by the editors September 25, 1996; accepted for publication (in revised form) December 4, 1998; published electronically April 29, 1999.

http://www.siam.org/journals/sidma/12-2/30978.html

†Department of Mathematics and Statistics, Dalhousie University, Halifax, Nova Scotia, B3H 3J5 Canada (janssen@mscs.dal.ca).

‡Centre of Discrete and Applicable Mathematics, London School of Economics, Houghton Street, London WC2A 2AE, England (kyri@cdam.lse.ac.uk).

A *k-stable set* of G is a stable set of cardinality at most k. By letting $\mathcal{J}$ be the family of k-stable sets of G, we have the analogous notion of a coloring (fractional or integral) by k-stable sets. The corresponding values of optimal colorings will be denoted by $\chi_k(G, \mathbf{c})$ and $\eta_k(G, \mathbf{c})$. The convex hull of incidence vectors of k-stable sets of G will be referred to as the k-stable set polytope of G. This is a polytope of full dimension in $\mathrm{Q}^{V(G)}$, since all unit vectors and the zero vector correspond to k-stable sets of G.

There are two related problems we deal with in this paper. The first involves "slicing off" the stable set polytope of a graph G with the inequality $\sum_{v \in V(G)} x(v) \leq k$, for some positive integer k, and studying the remaining polytope. The integral points of this polytope correspond to the k-stable sets of G, and we are interested in the inequalities that define their convex hull.

For $k = 2$ we give a complete inequality description of the k-stable set polytope of any graph. For $k = 3$ we give an analogous result for the k-stable set polytope of bipartite graphs. In either case, the system described is totally dual integral and each inequality is facet defining.

The second problem involves the study, for any $\mathbf{c} \in \mathrm{Z}_+^{V(G)}$, of the relationship between integral and fractional colorings by k-stable sets. A description of the k-stable set polytope of a graph G gives a formula for $\eta_k(G, \mathbf{c})$, the value of an optimal fractional coloring of $(G, \mathbf{c})$ by k-stable sets for any $\mathbf{c} \in \mathrm{Z}_+^{V(G)}$. This formula can be used as a starting point for studying (integral) colorings of $(G, \mathbf{c})$ by k-stable sets. In particular, when $\chi_k(G, \mathbf{c}) = \lceil \eta_k(G, \mathbf{c}) \rceil$, the corresponding polytope yields necessary and sufficient conditions on the colorability of G by k-stable sets. In general, it is desirable to obtain upper bounds on $\chi_k(G, \mathbf{c})$ expressed as functions of $\eta_k(G, \mathbf{c})$.

For $k = 2$ we show that $\chi_k(G, \mathbf{c}) \leq \eta_k(G, \mathbf{c}) + \frac{\eta(G)}{2}$, where $\eta(G) = \eta(G, \mathbf{1})$. For $k = 3$ we show that $\chi_k(G, \mathbf{c}) \leq \lceil \eta_k(G, \mathbf{c}) \rceil + 1$ when G is bipartite. In terms of polytopes, and in view of the results of Baum and Trotter, Jr. [2], the first inequality shows that for every vector $\mathbf{x}$ in the 2-stable set polytope of G and every $p \in \mathrm{Z}_+$, $p\mathbf{x}$ can be written as the sum of $\lceil \eta_k(G, \mathbf{c}) + \frac{\eta(H)}{2} \rceil$ incidence vectors of 2-stable sets. An analogous statement holds also for the second result.

Colorings by k-stable sets, or, as they are better known, k-bounded node colorings, arise in chromatic scheduling problems when the number of rooms, machines, or other resources is limited (see [11]). Chen and Lih [4] have established a formula for the k-bounded chromatic number of an unweighted tree. Lower bounds, upper bounds, and complexity results on k-bounded colorings are given by Hansen, Hertz, and Kuplinsky [7].

For colorings in general, there is no relationship between the fractional and integral chromatic number of a graph. In [12] a family of graphs G is given such that $\eta(G) \to 2$, whereas $\chi(G) \to \infty$ as $|V(G)| \to \infty$. However, for special classes of graphs, relationships between fractional and integral colorings have been established even for weighted graphs (see [10]). Colorings of line graphs (or edge colorings) in particular have attracted great attention. Several high profile conjectures attempt to link fractional and integral colorings in these graphs, but so far the results have been limited. We refer the reader to [9] and [8] for further explanations on the problems involved.

The stable set polytope of graphs G with stability number at most 2 has been described by Cook [5]. A proof of this result is also given in [15]. The stable set polytope in general has been studied extensively. The reader is referred to [16] and [1], where links to the corresponding literature can be found.

2. Definitions and preliminaries. The graph-theoretical and polyhedral concepts not defined here can be found in Bondy and Murty [3] and Schrijver [14], respectively.

A *graph* G is an ordered pair (V,E), consisting of a *node set* $V(G)$ and an *edge set* $E(G)$. The edges of G form a subset of $\{\{u,v\} : u,v \in V(G), u \neq v\}$. An edge $\{u,v\}$ is simply denoted by uv. The *complement* of G, denoted $\bar{G}$, is the graph $(V(G), \{uv : u,v \in V(G), uv \notin E(G)\})$. A *matching* of G is a set of independent edges. A *clique* of G is a subset of mutually adjacent nodes of G. For $v \in V(G)$, $N(v)$ denotes the set $\{v \in V(G) : uv \in E(G)\}$, and for $T \subseteq V(G)$, $N(T)$ denotes the set $\bigcup_{v \in T} N(v) \backslash T$. For a subset F of $V(G)$, the subgraph of G *induced* by F, denoted $G[F]$, is the graph $(F, \{uv \in E(G) : u,v \in F\})$. We will use the following well-known result.

THEOREM 2.1 (Hall's theorem). *If G is a bipartite graph with partition (A,B), then G has a matching of cardinality $|A|$ if and only if for every $T \subseteq A$, $|N(T)| \geq |T|$.*

Let $\mathbf{c} \in \mathrm{Z}_+^{V(G)}$ and $v \in V(G)$. We denote by $\alpha(G,\mathbf{c})$ the number $\max\{c(S) : S \text{ is a stable set of } G\}$, and $\alpha(G)$ $(= \alpha(G,\mathbf{1}))$ is the *stability number* of G. Any stable set that yields this maximum is referred to as a *maximum weight* stable set of $(G,\mathbf{c})$. We use the analogous notion for k-stable sets and the corresponding maximum is denoted by $\alpha_k(G,\mathbf{c})$. Denote by $\delta(v)$ the edges of G that contain v. An *edge covering* (respectively, a *b-matching*) of $(G,\mathbf{c})$ is collection of edges F such that for every $v \in V(G)$, $|\delta(v) \cap F| \geq c(v)$ (respectively, $|\delta(v) \cap F| \leq c(v)$). In terms of vectors, an edge covering of $(G,\mathbf{c})$ is any point in

$$\{\mathbf{x} \in \mathrm{Z}_+^{E(G)} : x(\delta(v)) \geq c(v), v \in V(G)\},$$

and a b-matching of $(G,\mathbf{c})$ is any point in

$$\{\mathbf{x} \in \mathrm{Z}_+^{E(G)} : x(\delta(v)) \leq c(v), v \in V(G)\}.$$

In both case, by letting $\mathbf{x} \in \mathrm{Q}_+^{E(G)}$ we obtain the corresponding notions of *fractional edge coverings* and *fractional b-matchings* of $(G,\mathbf{c})$.

Let $\mathbf{c}$ be a vector indexed by $V(G)$, F a subset of $V(G)$. The *restriction of* $\mathbf{c}$ *to* F is the $|F|$-dimensional vector whose components correspond to the components of $\mathbf{c}$ indexed by F. We denote by $c(v)$, $v \in V(G)$, the component of $\mathbf{c}$ indexed by v and by $c(F)$ the number $\sum_{v \in F} c(v)$. The *incidence vector* of F, denoted χ^F, is a vector indexed by $V(G)$, with a component equal to one if the corresponding node belongs to F and is equal to zero otherwise. The *support* of $\mathbf{c}$ are those nodes of G that index nonzero components of $\mathbf{c}$.

To avoid confusion, we digress from the above notation when a vector $\mathbf{y}$ is indexed by a family $\mathcal{S}$ of subsets of $V(G)$. In this case, we use y_S to denote the component of $\mathbf{y}$ indexed by $S \in \mathcal{S}$.

An inequality $\mathbf{c} \cdot \mathbf{x} \leq \delta$ is *implied* by a system of linear inequalities $\mathbf{Ax} \leq \mathbf{b}$ if every $\mathbf{x}$ that satisfies the system also satisfies $\mathbf{c} \cdot \mathbf{x} \leq \delta$.

THEOREM 2.2 (Farkas's lemma). *Suppose that the inequality system $\mathbf{Ax} \leq \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$ has a solution. Then $\mathbf{c} \cdot \mathbf{x} \leq \delta$ is implied by this system if and only if there exists a row vector $\lambda \geq \mathbf{0}$ such that $\lambda \mathbf{A} \geq \mathbf{c}$ and $\lambda \cdot \mathbf{b} \leq \delta$.*

In terms of k-stable sets, Farkas's lemma tells us that $\mathbf{Ax} \leq \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$ defines the k-stable set polytope of a graph G if and only if for every $\mathbf{c} \in \mathrm{Z}_+^{V(G)}$, $\mathbf{c} \cdot \mathbf{x} \leq \alpha_k(G,\mathbf{c})$ is implied by $\mathbf{Ax} \leq \mathbf{b}$. When λ in the above theorem can be chosen to be integral, the system $\mathbf{Ax} \leq \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$ is said to be *totally dual integral*.

Given a polytope Q, an inequality $\mathbf{a}\cdot\mathbf{x}\leq\beta$ is *facet defining* if it is valid for all points of Q and the set $\{\mathbf{x}\in Q:\mathbf{a}\cdot\mathbf{x}=\beta\}$ is a facet of Q.

Padberg [13] has introduced a procedure, called *sequential lifting*, which can be used to build facet defining inequalities for the k-stable set polytope of a graph G from those for induced subgraphs of G. Let $X\subseteq V(G)$, and let $\mathbf{a}\cdot\mathbf{x}\leq\beta$ be a facet defining inequality for the k-stable set of $G[X]$. Let v be any node of $V(G)\backslash X$, and let $\pi=\beta-\max\{\mathbf{a}\cdot\chi^S : S \text{ is a } (k-1)\text{-stable set of } G[X\backslash N(v)]\}$. Then the inequality $\pi x(v)+\mathbf{a}\cdot\mathbf{x}\leq\beta$ is facet defining for the k-stable set polytope of $G[X\cup\{v\}]$. A *lift* of $\mathbf{a}\cdot\mathbf{x}\leq\beta$ (in G) is any inequality obtained by a sequential application of this procedure.

3. Polytopes. In this section we describe the 2-stable set polytope of graphs in general and the 3-stable set of bipartite graphs. We begin with 2-stable sets.

The following theorem includes results of Cook and Shepherd and its proof can be found in [15]. For a subset K of $V(G)$, we denote by $\tilde{N}(K)$ the set $\bigcap_{v\in K}N(v)\backslash K$.

THEOREM 3.1. *For any graph G with $\alpha(G)\leq 2$, the following system describes the stable set polytope of G and is totally dual integral, and each inequality is facet defining.*

(0) $x(v)\geq 0$ *for all* $v\in V(G)$.

(1) $x(K)\leq 1$ *for all maximal cliques* K.

(2) $2x(K)+x(\tilde{N}(K))\leq 2$ *for each clique K such that none of the connected components of the complement of $G[\tilde{N}(K)]$ is a bipartite graph.*

(3) $x(V(G))\leq 2$ *if no connected component of the complement of G is bipartite.*

Given the above theorem, it is rather easy to describe the 2-stable set of any graph.

DEFINITION 3.2. *For a graph G, the system $\mathbf{A}_2\mathbf{x}\leq\mathbf{b}$ consists of the inequalities* (1)–(3) *of Theorem* 3.1 *as well as the following:*

(4) $x(A)+2x(K)+x(\tilde{N}(K)\backslash A)\leq 2$, *for each set A such that every maximal stable set in $G[A]$ has size at least* 2 *and $G[A]$ has a stable set of size at least* 3, *and for every clique K maximal in $\tilde{N}(A)$.*

Note that all inequalities of type (4) can be obtained from the inequality $x(S)\leq 2$, where S is a stable set of size 3, by sequential lifting. If $S=\{v_1,v_2,v_3\}$, the vectors $\chi^{\{v_1,v_2\}}$, $\chi^{\{v_1,v_3\}}$, $\chi^{\{v_2,v_3\}}$ are affinely independent, and they satisfy $x(S)\leq 2$ with equality. So the inequality $x(S)\leq 2$ is facet defining for $G[S]$. When we lift this inequality sequentially to all vertices of a set A that has the property that all stable sets of $G[A]$ have size at least 2, then all vertices of A will have coefficient 1, and we obtain the inequality $x(A)\leq 2$, facet defining for $G[A]$. If we then lift this inequality to a vertex v in $\tilde{N}(A)$, this vertex will get coefficient 2, since the maximal stable set in $G[A\backslash N(v)]$ is of size 0. The same is true for all vertices in a maximal clique K in $\tilde{N}(A)$ that includes v. So we obtain the inequality $x(A)+2x(K)\leq 2$. When we lift this inequality to the rest of the graph, any vertex v will get coefficient 0 if $K\not\subseteq N(v)$ and coefficient 1 otherwise. So we obtain the inequality $x(A)+2x(K)+x(\tilde{N}(K)\backslash A)\leq 2$, an inequality of type (4).

Note also that the inequality $x(V(G))\leq 2$ is implied by the inequality system whenever $\alpha(G)\geq 3$. If no connected component of the complement of G is bipartite, then this inequality is included in type (3). Suppose then that a connected component of the complement of G is bipartite. This means that G contains two cliques, K_1 and K_2, such that $\tilde{N}(K_1)=\tilde{N}(K_2)=V(G)\backslash(K_1\cup K_2)$. If $V(G)\backslash(K_1\cup K_2)$ contains a stable set of size at least 3, then the inequality $x(V)\leq 2$ is implied by two inequalities of type (4), one with $K_1\subseteq K$, one with $K_2\subseteq K$, and both with $A\subseteq V(G)\backslash(K_1\cup K_2)$.

If $V(G)\backslash(K_1 \cup K_2)$ does not contain a stable set of size at least 3, then $\alpha(G) = 2$.

THEOREM 3.3. *For a graph G, the system $\mathbf{A}_2\mathbf{x} \leq \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$ describes the 2-stable set polytope of G and is totally dual integral, and each of the inequalities is facet defining.*

Proof. In view of Theorem 3.1, for each inequality of types (1)–(3) with support X, there exists a collection of $|X|$ affinely independent stable sets which satisfy it with equality and which are subsets of X of size at most 2. Therefore, each of these inequalities is facet defining for the 2-stable set polytope of $G[X]$. When the inequalities are lifted to the rest of $V(G)$, they remain facet defining.

For the inequalities of type (4), it is already noted that they can be obtained by lifting the facet defining inequality $x(S) \leq 3$, where S is a stable set of size 3. Therefore, all inequalities of the system $\mathbf{A}_2\mathbf{x} \leq \mathbf{b}$ are facet defining.

For the rest of the proof, let $\mathbf{c}$ be a nonnegative integral vector indexed by the nodes of G. We show that there exists an integral, nonnegative row vector λ indexed by the rows of $\mathbf{A}_2$ such that $\lambda\mathbf{A} \geq \mathbf{c}$ and $\lambda \cdot \mathbf{b} \leq \alpha_2(G, \mathbf{c})$.

We apply induction on $\alpha_2(G, \mathbf{c})$. If $\alpha_2(G, \mathbf{c}) = 0$, then $\mathbf{c} = \mathbf{0}$, because every vertex is by itself a stable set. So we can choose $\lambda = \mathbf{0}$. Next, we assume that the theorem holds for any $\mathbf{c}'$ with $\alpha_2(G, \mathbf{c}') < \alpha_2(G, \mathbf{c})$. Furthermore, we may assume without loss of generality that $c(v) > 0$ for all $v \in V(G)$, since nodes with zero weight can be deleted from the graph. (This is because any of the above inequalities can be lifted to an inequality in G that contains v and is of one of the specified types.)

Note that it suffices to find an inequality $\mathbf{a} \cdot \mathbf{x} \leq \beta$ of the system $\mathbf{A}_2\mathbf{x} \leq \mathbf{b}$ such that $\alpha_2(G, \mathbf{c} - \mathbf{a}) \leq \alpha_2(G, \mathbf{c}) - \beta$ and $\mathbf{c} - \mathbf{a} \geq \mathbf{0}$. Then, by induction, there exists a nonnegative integral vector λ' such that $\lambda'\mathbf{A}_2 \geq \mathbf{c} - \mathbf{a}$ and $\lambda' \cdot \mathbf{b} \leq \alpha_2(G, \mathbf{c} - \mathbf{a})$. Let λ be obtained from λ' by simply increasing the coordinate of λ' indexed by this inequality by 1. This yields $\lambda\mathbf{A}_2 \geq \mathbf{c}$ and $\lambda \cdot \mathbf{b} \leq \alpha_2(G, \mathbf{c} - \mathbf{a}) + \beta \leq \alpha_2(G, \mathbf{c})$.

If $\alpha(G) \leq 2$, then we are done by Theorem 3.1. Thus we assume that $\alpha(G) \geq 3$. If every maximum weight 2-stable set has cardinality 2, then let $\mathbf{a} \cdot \mathbf{x} \leq \beta$ be the inequality $x(V(G)) \leq 2$, if either this inequality or the weighted sum of the two inequalities of type (4) that implies the inequality $x(V(G)) \leq 2$ are included in the system (see the note following the statement of the Theorem). In this case it is immediate that $\alpha_2(G, \mathbf{c} - \mathbf{a}) \leq \alpha_2(G, \mathbf{c}) - \beta$. Otherwise, there exists a node v such that $c(v) = \alpha_2(G, \mathbf{c})$ (and $\alpha_2(G, \mathbf{c}) \geq 2$), and thus $N(v) = V(G)\backslash\{v\}$. Let K be a clique of G that contains v such that $\tilde{N}(K) = V(G)\backslash K$ and K is maximal with this property. Let $\mathbf{a} \cdot \mathbf{x} \leq \beta$ be the inequality $2x(K) + x(\tilde{N}(K)) \leq 2$. Because of the maximality condition on K, all stable sets in $G[V(G)\backslash K]$ have size at least 2, so this inequality is of type (4), with $A = V(G)\backslash K$. Note that every maximum weight 2-stable set of G consists of either an element of K or two elements of $\tilde{N}(K)$. Thus, since $c(v) \geq 2$ and because K must include all vertices v such that $c(v) = \alpha_2(G, \mathbf{c})$, $\alpha_2(G, \mathbf{c} - \mathbf{a}) \leq \alpha_2(G, \mathbf{c}) - \beta$. The proof is complete. □

Before we can adequately describe the facet defining inequalities of the 3-stable set polytope of bipartite graphs, we need a few definitions.

DEFINITION 3.4. *A 2-star is a graph whose node set can be partitioned into two sets $T = \{v_1, v_2\}$ and B such that T is a stable set and $B = N(T)$. With any given 2-star we identify the sets $L = N(v_1)\backslash N(v_2)$, $M = N(v_1) \cap N(v_2)$, and $R = N(v_2)\backslash N(v_1)$. A 2-star is full if $|N(v_1)|, |N(v_2)| \geq 3$ and $|L|, |R| \geq 2$.*

DEFINITION 3.5. *For a bipartite graph G, the inequality system $\mathbf{A}_3\mathbf{x} \leq \mathbf{b}$ consists of the following inequalities:*

(1) $x(K) \leq 1$ *for all maximal cliques K of G (i.e., edges and isolated nodes);*

(2) $x(V(G)) \leq 3$ *if both parts of a bipartition of* G *have at least four elements;*
(3) *the lift of* $x(S) \leq 3$ *for each stable set* S *of size at least* 4*;*
(4) $2x(T) + 2x(u) + x(B\backslash\{u\}) \leq 4$ *for each full* 2*-star with* $M \neq \emptyset$ *and for each node* $u \in M$*;*
(5) $4x(v_1) + 2x(v_2) + 2x(N(v_1)) + x(R) \leq 6$ *for each full* 2*-star with* $|R| \geq 3$ *and* $|N(v_1)| \geq 4$ *if* $M \neq \emptyset$ *and each ordering* (v_1, v_2) *of* T*.*

When the inequality $x(S) \leq 3$, where S is a stable set of size at least 4, is lifted to the rest of the graph, and a vertex v is encountered that is adjacent to all previously lifted vertices, then v will get coefficient 3 in the inequality. All other vertices will then have coefficient 1 if they are adjacent to v, and 0 if this is not the case. If a vertex v is adjacent to all previously lifted vertices except one single vertex or two adjacent vertices, then v will get coefficient 2. All other vertices will get coefficient 1 if they are adjacent to v, or 0 if this is not the case. Otherwise, the inequality will be lifted to $x(V(G)) \leq 3$. So the type (3) inequalities are of the following form:

(3*a*) $3x(v) + x(N(v)) \leq 3$ for all $v \in V(G)$ with $|N(v)| \geq 4$;
(3*b*) $2x(v) + x(N(v)) + x(K) \leq 3$ for all $v \in V(G)$ and each clique K with no element in common with $N(v) \cup \{v\}$;
(3*c*) $x(V(G)) \leq 3$ if G cannot be obtained from $K_{m,n}$, $m \geq 1$, $n \geq 4$, by deleting a (possibly empty) matching.

THEOREM 3.6. *For a bipartite graph* G*, the normalized version of the system* $\mathbf{A}_3\mathbf{x} \leq \mathbf{b}$ *(i.e., where all inequalities have been divided by the right-hand side),* $\mathbf{x} \geq \mathbf{0}$ *is totally dual integral and describes the* 3*-stable set polytope of* G*, and each inequality is facet defining.*

Proof. We begin by showing that each of the inequalities given is facet defining. This is done by exhibiting, in each case, the appropriate sets of affinely independent vectors that satisfy the given inequality with equality. For convenience, we will say that we find affinely independent stable sets, instead of affinely independent incidence vectors of stable sets. The following fact can be proved by using the notion of lifting as in the proof of Theorem 3.3.

STEP 3.6.1. *If* S*,* $|S| \geq 4$ *(respectively,* $|S| \geq 3$*), is a stable set of* G*, then it has* $|S|$ *affinely independent subsets of size* 3 *(respectively,* 2*).*

The inequalities $\mathbf{x} \geq \mathbf{0}$ and those of type (1) are well known to be facet defining. From (3.6.1) and the lifting procedure, it is immediate that the inequalities of type (3) are facet defining. The same lemma may also be applied to each part of a bipartition of G and thus the inequality of type (2) is facet defining as well. Consider now an inequality of type (4). By (3.6.1), $B\backslash\{u\}$ has $|B| - 1$ affinely independent stable sets of two nodes and thus, by including u to all these sets, B has $|B| - 1$ such sets of three nodes. Adding also the sets $\{v_1, v_2\}$, $\{v_1, u_3, u_4\}$, and $\{v_2, u_1, u_2\}$, where $u_1, u_2 \in L$ and $u_3, u_4 \in R$, we obtain a collection of $|B| + |T|$ affinely independent 3-stable sets that satisfy the given inequality with equality.

Finally, consider an inequality of type (5), and assume that $|N(v_1)| \geq 4$. $N(v_1)$ contains $|N(v_1)|$ affinely independent stable sets of size 3. Also, R has $|R|$ affinely independent stable sets of size 2. By including v_1 to these sets, we have that $R \cup \{v_1\}$ has $|R|$ affinely independent stable sets of size 3. Adding the sets $\{v_1, v_2\}$ and $\{v_2, u_1, u_2\}$, where $u_1, u_2 \in L$, we obtain the required collection of affinely independent stable sets. When $|N(v_1)| = 3$ instead of $N(v_1)$, we use $L \cup \{v_2\}$ to obtain $|L|$ affinely independent 3-stable sets.

We continue with the rest of the proof. Let $\mathbf{c}$ be a nonnegative integral vector indexed by the nodes of G. We show that there exists an integral, nonnegative row vector λ indexed by the rows of $\mathbf{A}_3$ such that $\lambda\mathbf{A_3'} \geq \mathbf{c}$ and $\lambda \cdot \mathbf{1} \leq \alpha_3(G, \mathbf{c})$, where $A_3'\mathbf{x} \leq \mathbf{1}$ is the normalized version of the system $\mathbf{A}_3\mathbf{x} \leq \mathbf{b}$.

The proof is by induction on $\alpha_3(G, \mathbf{c})$. If $\alpha_3(G, \mathbf{c}) = 0$, then $\mathbf{c} = \mathbf{0}$, so we can take $\lambda = \mathbf{0}$. Next, we assume that the lemma holds for any $\mathbf{c}'$ with $\alpha_3(G, \mathbf{c}') < \alpha_3(G, \mathbf{c})$. Furthermore, we may assume without loss of generality that $c(v) > 0$ for all $v \in V(G)$. (This is because any of the above inequalities for $G - v$, $v \in V(G)$, can be lifted to an inequality of G that contains v and is of one of the specified types.) As in the proof of Theorem 3.3, it suffices to find an inequality $\mathbf{a} \cdot \mathbf{x} \leq \beta$ which is the sum (with the appropriate coefficients) of inequalities of $\mathbf{A}_3\mathbf{x} \leq \mathbf{b}$ and such that $\alpha_3(G, \mathbf{c} - \mathbf{a}) \leq \alpha_3(G, \mathbf{c}) - \beta$ and $\mathbf{c} - \mathbf{a} \geq \mathbf{0}$.

If every stable set of G is of size at most 3, then the stable set and 3-stable set polytopes of G coincide, and it is well known that the former is given by the inequalities of types (0) and (1). Otherwise, if every maximum weight 3-stable set of G is of size exactly 3 and for all $v \in V(G)$, $c(v) < \alpha_3(G, \mathbf{c}) - 1$, then the required inequality is $x(V(G)) \leq 3$. Clearly, $\alpha_3(G, \mathbf{c} - \chi^{V(G)}) = \alpha_3(G, \mathbf{c}) - 3$. If the inequality $x(V(G)) \leq 3$ is not of type (2) or (3c), then by the definition of $\mathbf{A}_3 x \leq b$, each bipartition of G has a part with less than four nodes, and G can be obtained from $K_{m,n}$, $m \geq 1$, $n \geq 4$ by deleting a matching. If G can be obtained from $K_{3,n}$, $n \geq 4$, by deleting a matching, then $x(V(G)) \leq 3$ is the sum (with coefficients $\frac{1}{3}$) of three inequalities of type (3b). If G can be obtained from $K_{2,n}$ or $K_{1,n}$ by deleting a matching, then $x(V(G)$ is implied by an inequality of type (3b) or (3a).

Thus we may assume that G has a maximal 3-stable set T of size 1 or 2 and if $|T| = 2$, then T is of maximum weight. Let $B = N(T)$. By increasing the weight of components of $\mathbf{c}$ that correspond to vertices that are not part of maximum weight stable sets, if necessary, we may thus assume the following.

STEP 3.6.2. *Every node of G belongs to a maximum weight 3-stable set of* $(G, \mathbf{c})$.

We now distinguish two cases.

Case 1. G is not a full 2-star.

If T has only one element v, then every 3-stable set of maximum weight consists of either v or three nodes in $N(v)$. Thus $3x(v) + x(N(v)) \leq 3$ is the required inequality, since by (3.6.2), $c(v) = \alpha_3(G, \mathbf{c}) \geq 3$.

Thus we may assume that G is a 2-star, although not a full one, and since $\alpha_3(G) > 3$, $|B| \geq 4$. Let v_1 and v_2 be the elements of T, ordered so that $|N(v_2)| \leq |N(v_1)|$. It can be argued from the fact that $c(v_1) + c(v_2) = \alpha_3(G, \mathbf{c})$ that $c(v_1) \geq 2$. Let $\mathbf{a} \cdot \mathbf{x} \leq \beta$. be the inequality $2x(v_1) + x(N(v_1)) + x(K) \leq 3$ (of type (3b)), where K is a maximal clique in $\{v_2\} \cup R$. Every maximum weight 3-stable set of $(G, \mathbf{c})$ is also of maximum weight in $(G, \mathbf{a})$ and $\mathbf{c} - \mathbf{a} \geq \mathbf{0}$. Thus $\alpha_3(G, \mathbf{c} - \mathbf{a}) \leq \alpha_3(G, \mathbf{c}) - \beta$, as required.

Case 2. G is a full 2-star.

We adopt the notation introduced in Definition 3.4. To proceed, we need to identify certain nodes of G and introduce a simple lemma. Let t_1 and t_2 be the weights of v_1 and v_2, and let $m_1, m_2, \ldots$ be the weights of the nodes of M, $\ell_1, \ell_2, \ldots$ the weights of the nodes of L, and $r_1, r_2, \ldots$ the weights of the nodes of R, all in decreasing order.

STEP 3.6.3. *There is no 3-stable set S of maximum weight such that $S \cap L \neq \emptyset$, $S \cap R \neq \emptyset$, and $S \cap M = \emptyset$.*

This can be deduced from the fact that $r_1 + r_2 \leq t_2$ and $\ell_1 + \ell_2 \leq t_1$ (because $t_1 + t_2 = \alpha_3(G, \mathbf{c})$), and thus both $\ell_1 + r_1 + r_2$ and $\ell_1 + \ell_2 + r_1$ are less than $t_1 + t_2$.

We consider three subcases.

The simplest case arises when no maximum weight 3-stable set of G is a subset of B. Let $\mathbf{a} \cdot \mathbf{x} \le \beta$ denote the inequality $2x(T) + x(B) \le 4$. Since $t_1 + t_2 = \alpha_2(G, \mathbf{c})$, $t_1 \ge \ell_1 + \ell_2 \ge 2$, $t_2 \ge r_1 + r_2 \ge 2$ and thus $\mathbf{c} - \mathbf{a} \ge 0$. In addition, $\alpha_3(G, \mathbf{c} - \mathbf{a}) \le \alpha_3(G, \mathbf{c}) - \beta$. Now if $M \ne \emptyset$, then $\mathbf{a} \cdot \mathbf{x} \le \beta$ is implied by an inequality of type (4); if $M = \emptyset$, then it is implied by the two inequalities of type (5) (with coefficients $\frac{1}{3}$) that correspond to the two orderings of T.

Next, suppose there is a 3-stable set of maximum weight that contains nodes from each of L, R, and M—in other words, $r_1 + \ell_1 + m_1 = t_1 + t_2$. Our analysis depends on whether or not $|M| \ge 2$ and $m_1 = m_2$.

If $|M| \ge 2$ and $m_1 = m_2$, then let $\mathbf{a} \cdot \mathbf{x} \le \beta$ be the inequality $3x(T) + x(B) + x(\{u_\ell, u_r\})$, where u_ℓ and u_r are the nodes of weight ℓ_1 and r_1, respectively. Now by (3.6.3), $r_1 + \ell_1 + l_2 < \alpha_3(G, \mathbf{c}) = \ell_1 + m_1 + r_1$, so $\ell_2 < m_1$, and thus $\ell_1 + m_1 + m_2 > \ell_1 + \ell_2 + m_1$. By symmetry, $r_1 + m_1 + m_2 > r_1 + r_2 + m_2$. Therefore, u_ℓ and u_r are the only nodes from L and R, respectively, that are contained in maximum weight stable sets. Also, since $\ell_1 > \ell_2 \ge 1$ and $r_1 > r_2 \ge 1$, $\ell_1, r_1 \ge 2$, and since $\ell_1 + \ell_2 \le t_1$ and $r_1 + r_2 \le t_2$, we have that $t_1, t_2 \ge 3$. It follows that $\mathbf{a} \cdot \mathbf{x} \le \mathbf{b}$ has the property that all the maximum weight 3-stable sets of $(G, \mathbf{c})$ are also maximum weight stable sets of $(G, \mathbf{a})$, and thus $\alpha_3(G(\mathbf{c} - \mathbf{a})) \le \alpha_3(G, \mathbf{c}) - \beta$, since $\mathbf{c} - \mathbf{a} \ge \mathbf{0}$. In addition, $\mathbf{a} \cdot \mathbf{x} \le \beta$ is the sum of two inequalities of type (3b).

If $|M| = 1$ or $m_1 > m_2$, then all maximum weight 3-stable sets in B have to include the node $u \in M$ of weight m_1. Also, since $\ell_1 + \ell_2 \le t_2$, $r_1 + r_2 \le t_1$, and because, by (3.6.3), $r_1 + r_2 + \ell_1 \le \alpha_3(G, \mathbf{c}) = r_1 + m_1 + \ell_1$ (so $m_1 > r_2$), we have that t_1, t_2, $m_1 \ge 2$. Thus by taking $\mathbf{a} \cdot \mathbf{x} \le \beta$ to be the type (4) inequality $2x(T) + 2x(u) + x(B \backslash \{u\}) \le 4$, we have $\mathbf{c} - \mathbf{a} \ge \mathbf{0}$ and $\alpha_3(G, \mathbf{c} - \mathbf{a}) \le \alpha_3(G, \mathbf{c}) - \beta$.

To conclude, we assume that every maximum weight 3-stable set that is contained in B is also contained in, say, $N(v_1)$. The inequality $\mathbf{a} \cdot \mathbf{x} \le \beta$ of the form $4x(v_1) + 2x(v_2) + 2x(N(v_1)) + x(R) \le 6$ has the property that every maximum weight stable set of $(G, \mathbf{c})$ is also a maximum weight stable set of $(G, \mathbf{a})$. If $|R| = 2$, this inequality is the sum of two inequalities of type (3b); if $M \ne \emptyset$ and $|N(v_1)| = 3$, it is the sum of two inequalities of type (1) and one inequality of type (4); otherwise it is an inequality of type (5). In either case we are done, provided that $t_1 \ge 4$, $t_2 \ge 2$, and $c(u) \ge 2$ for all $u \in N(v_1)$.

Since $t_1 + t_2 = \alpha_3(G, \mathbf{c})$, $t_1 + t_2 \ge t_1 + r_1 + r_2$ and thus $t_2 \ge 2$. Next we show that $c(u) \ge 2$ for all $u \in N(v_1)$. If $u \in M$, then by (3.6.2), there is a 3-stable set $S = \{u, w_1, w_2\}$ of maximum weight that contains u. Since $u \in M$, $S \subseteq B$. By the assumption that every maximum weight stable set is contained in $N(v_1)$, $c(w_1) + c(w_2) + r_1 < \alpha_3(G, \mathbf{c}) = c(u) + c(w_1) + c(w_2)$. Thus $c(u) \ge 2$. Now if $u \in L$ we may assume that $c(u)$ is minimal. Suppose that every maximum weight stable set that contains u also contains v_2. (If this is not so, we are done by the same reasoning as for $u \in M$.) Then $c(u) \ge \ell_2$, and since $c(u)$ is minimal, $c(u) = \ell_2$ and $\ell_1 + 2\ell_2 < \alpha_3(G, \mathbf{c})$. By assumption, there is a maximum weight stable set S' contained in B, but by the above, it does not contain a node with weight ℓ_2. Thus $|M| \ge 2$ and $m_1 + m_2 + \ell_1 \le \alpha_3(G, \mathbf{c}) = \ell_1 + \ell_2 + t_2$. But, by (3.6.2), $r_1 + r_2 = t_2 \ge m_1 + m_2 - \ell_2$ and thus $\ell_2 \ge m_1 + m_2 - r_1 - r_2$. But we saw that $m_i > r_1$ (so $m_i - r_1 \ge 1$) for all i, so $\ell_2 \ge 2$. Thus for all $u \in N(v_1)$, $c(u) \ge 2$. Finally, since $t_1 + t_2 = \alpha_3(G, \mathbf{c}) \ge t_2 + \ell_1 + \ell_2 \ge t_2 + 4$, we have $t_1 \ge 4$, as required. □

Parenthetically, we remark that the ideas behind the inequalities of types (4) and (5) of Definition 3.5 can be extended to construct facet defining inequalities for the stable set polytopes of graphs. Let H be a graph such that $x(V(H)) \le \alpha(H)$ is a facet defining inequality for the stable set polytope of H and H has two stable sets S_1, S_2

with $|S_1| = |S_2| = \alpha(H)$ and $S_1 \cap S_2 = \emptyset$. Let G be a graph obtained from H by adding three new nodes v_1, v_2, u and for $i = 1, 2$ the edges $\{v_i u\} \cup \{v_i v : v \in V(H) \backslash S_i\}$. Let $\alpha = \alpha(G)$. Then the inequality

$$(\alpha - 1)x(\{v_1, v_2, u\}) + x(V(H)) \leq 2\alpha - 2$$

is facet defining for the stable set polytope of G. Alternatively, let H_1 and H_2 be two graphs such that the inequalities $x(V(H_1)) \leq \alpha(H_1)$ and $x(V(H_2)) \leq \alpha(H_1)$ are facet defining for the stable set polytope of H_1 and H_2, respectively. Let G be obtained from H_1 and H_2 by adding two new nodes v_1 and v_2 and the edges $\cup_{i=1}^{2} \{v_i v : v \in V(H_i)\} \cup \{vw : v \in V(H_1), w \in V(H_2)\}$. Let $\alpha = \alpha(G)$. Then the inequality

$$(\alpha(\alpha - 2) + 1)x(v_1) + (\alpha - 1)x(v_2) + (\alpha - 1)x(V(H_1)) + x(V(H_2)) \leq \alpha(\alpha - 1)$$

is facet defining for the stable set polytope of G. The proof in both cases is analogous to the one given in Theorem 3.6. (Examples illustrating both constructions are shown in Figure 3.1; the coefficients not shown are equal to 1 and the right-hand sides are, respectively, 6 and 12; in both cases $\alpha = 4$.) We note that the above constructions can be generalized to obtain many families of facet defining inequalities for the stable set polytopes of graphs, but this falls beyond the scope of the present paper.

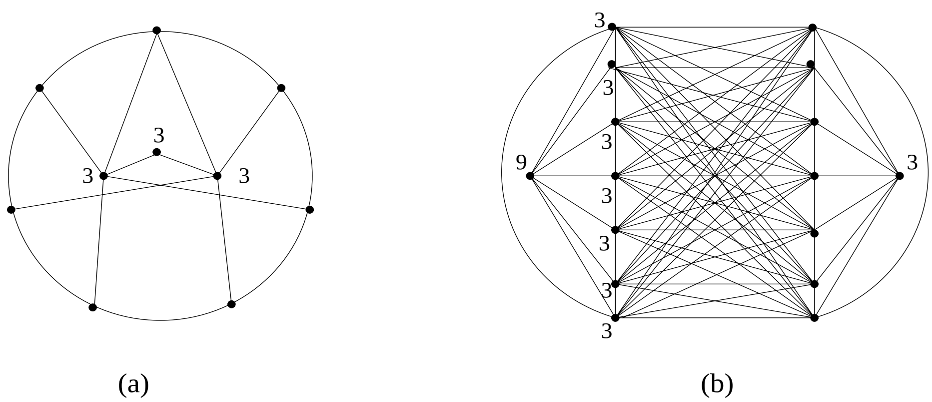


Fig. 3.1. *An illustration of facet defining inequalities for the stable set polytope.*

4. Colorings. We will now use the results of the previous section to obtain upper bounds on the value of integral colorings of the corresponding cases involved.

Note first that a description of the k-stable set polytope of a graph G provides necessary and sufficient conditions on the fractional colorability of $(G, \mathbf{c})$ by k-stable sets, where $\mathbf{c}$ is any vector in $Z_+^{V(G)}$. Indeed, by definition, $\mathbf{y}$ is a fractional coloring of $(G, \mathbf{c})$ of value r if and only if $\frac{1}{r}\mathbf{c}$ belongs to the k-stable set of G, i.e., if and only if $\mathbf{A}_k \mathbf{c} \leq r\mathbf{b}$, where $\mathbf{A}_k \mathbf{x} \leq \mathbf{b}$, $x \geq \mathbf{0}$ is an inequality description of the k-stable set of G. Thus,

$$\eta_k(G, \mathbf{c}) = \max\{\tfrac{1}{\beta}(\mathbf{a} \cdot \mathbf{c}) : \mathbf{a} \cdot \mathbf{x} \leq \beta \text{ is an inequality of } \mathbf{A}_k \mathbf{x} \leq \mathbf{b}\}.$$

We will use this min-max equality throughout. To begin, we introduce a simple lemma that explains how color classes from optimal fractional colorings intersect with the support of any given valid inequality of the k-stable set polytope of a graph.

LEMMA 4.1. *Let G be a graph and $\mathbf{c} \in Z_+^{V(G)}$. Let $\mathbf{a} \cdot \mathbf{x} \le \beta$, $\beta \ne 0$, $(\mathbf{a}, \beta)$ integral and nonnegative, be a valid inequality for the k-stable set polytope of G. If $\mathbf{a} \cdot \mathbf{c} = (\eta_k(G, \mathbf{c}) - t)\beta$, then in any optimal fractional coloring $\mathbf{y}$ of $(G, \mathbf{c})$ by k-stable sets, $\sum_{\{S \in \mathcal{S}: a(S) < \beta\}} y_S \le t\beta$.*

Proof. Let $\mathcal{S}_0$ (respectively, $\mathcal{S}_1$) consist of those k-stable sets $S \in \mathcal{S}$ such that $a(S) = \beta$ (respectively, $a(S) \le \beta - 1$). Let $p_0 = \sum_{S \in \mathcal{S}_0} y_S$ and $p_1 = \sum_{S \in \mathcal{S}_1} y_S$. Since $\sum_{S \in \mathcal{S}_0} y_S \chi^S + \sum_{S \in \mathcal{S}_1} y_S \chi^S \ge \mathbf{c}$,

$$\begin{aligned} \beta(\eta_k(G, \mathbf{c}) - t) &= \mathbf{a} \cdot \mathbf{c} \\ &\le \mathbf{a} \cdot (\textstyle\sum_{S \in \mathcal{S}_0} y_S \chi^S + \sum_{S \in \mathcal{S}_1} y_S \chi^S) \\ &= \textstyle\sum_{S \in \mathcal{S}_0} y_S a(S) + \sum_{S \in \mathcal{S}_1} y_S a(S) \\ &\le p_0 \beta + p_1(\beta - 1) \\ &= \eta_k(G, \mathbf{c}) - p_1, \end{aligned}$$

as required. □

Next we show that an upper bound on colorings by 2-stable sets can be obtained as the solution of two linear programs.

THEOREM 4.2. *For any graph G and any (strictly) positive vector $\mathbf{c} \in Z^{V(G)}$, $\chi_2(G, \mathbf{c}) \le \eta_2(G, \mathbf{c}) + \frac{\eta(G)}{2}$.*

Proof. We proceed by induction on $\eta_2(G, \mathbf{c})$. That is, we assume that the theorem holds for any graph G' and $\mathbf{c}' \in Z_+^{V(G')}$ with $\eta_2(G', \mathbf{c}') \le \eta_2(G, \mathbf{c}) - 1$. In the base case, G is the empty graph and the theorem holds trivially.

If the only inequality $\mathbf{a} \cdot \mathbf{x} \le \beta$ of $\mathbf{A}_2 \mathbf{x} \le \mathbf{b}$ for which $\mathbf{a} \cdot \mathbf{c} > (\eta_2(G, \mathbf{c}) - 1)\beta$ is the inequality $x(V(G)) \le 2$ (of type (3)), then let S be any 2-stable set of G of size 2. Let $\mathbf{d} = \mathbf{c} - \chi^S$. Then $\mathbf{A}_2 \mathbf{d} \le (\eta_2(G, \mathbf{c}) - 1)\mathbf{b}$, and consequently $\eta_2(G, \mathbf{d}) \le \eta_2(G, \mathbf{c}) - 1$. Let $G' = G - \{v \in V(G) : d(v) = 0\}$ and let $\mathbf{c}'$ be the restriction of $\mathbf{d}$ to $V(G'')$. By induction, $\chi_2(G', \mathbf{c}') \le \eta_2(G', \mathbf{c}') + \frac{\eta(G)}{2}$. Since any coloring of $(G, \mathbf{c}')$ can be augmented with S to form a coloring of $(G, \mathbf{c})$,

$$\chi_2(G, \mathbf{c}) \le \eta_2(G', \mathbf{c}') + \tfrac{\eta(G)}{2} + 1 \le \eta_2(G, \mathbf{c}) + \tfrac{\eta(G)}{2},$$

as required.

Thus we assume that there is an inequality $\mathbf{a} \cdot \mathbf{x} \le \beta$ of $\mathbf{A}_2 \mathbf{x} \le \mathbf{b}$ that is not of type (3) such that $\mathbf{a} \cdot \mathbf{c} > \eta_2(G, \mathbf{c}) - 1$. Choose this inequality so that for any other inequality $\mathbf{a}' \cdot \mathbf{x} \le \beta'$ that is not of type (3), $\frac{1}{\beta}(\mathbf{a} \cdot \mathbf{c}) \ge \frac{1}{\beta'}(\mathbf{a}' \cdot \mathbf{c})$. Let K be the clique involved in the definition of the support of $\mathbf{a}$. If the support of $\mathbf{a}$ is K, let G' be the empty graph and G'' be G itself. Otherwise, let G' be the subgraph $G[\tilde{N}(K)]$ of G and G'' the subgraph $G - \tilde{N}(K)$ of G. Let $\mathbf{c}'$ be the restriction of $\mathbf{c}$ to $V(G')$, and let $\mathbf{c}''$ be the restriction of $\mathbf{c}$ to $V(G'')$. We now use induction to color $(G', \mathbf{c}')$, whereas we color $(G'', \mathbf{c}'')$ explicitly.

Consider $(G', \mathbf{c}')$ and assume that G' is nonempty. By definition of the support of $\mathbf{a}$, every stable set in the graph $G[V(G') \cup K]$ that contains an element of K is of size 1. Also, by the choice of $\mathbf{a} \cdot \mathbf{x} \le \beta$, $G[V(G') \cup K]$ with $\mathbf{c}$ restricted to $V(G') \cup K$ has a fractional coloring by 2-stable sets of value $\frac{1}{\beta}(\mathbf{a} \cdot \mathbf{c})$. If we remove from this coloring all stable sets of size 1 that consist of a node of K, we are left with a fractional coloring of G' of value $\frac{1}{\beta}(\mathbf{a} \cdot \mathbf{c}) - c(K)$. Thus $\eta_2(G', \mathbf{c}') \le \frac{1}{\beta}(\mathbf{a} \cdot \mathbf{c}) - c(K)$. It follows, by induction, that

$$\chi_2(G', \mathbf{c}') \le \eta_2(G', \mathbf{c}') + \tfrac{\eta(G'}{2} \le \tfrac{1}{\beta}(\mathbf{a} \cdot \mathbf{c}) - c(K) + \tfrac{\eta(G')}{2}.$$

Consider $(G'', \mathbf{c}'')$. Let P be the node set of $G - (V(G') \cup K)$. Let $t = \eta_2(G, \mathbf{c}) - \frac{1}{\beta}(\mathbf{a} \cdot \mathbf{c})$. Note that $t = 0$ or $t = \frac{1}{2}$. We show that $\chi_2(G'', \mathbf{c}'') \leq c(K) + 2t$.

We construct a bipartite graph H with node set $A \cup B$ as follows. For each $v \in K$ (respectively, $v \in P$), let A_v (respectively, B_v) be a set of $c(v)$ nodes. Let $A = \cup_{v \in K} A_v$, $B = \cup_{v \in P} B_v$. To define the edges of H, consider an optimal fractional coloring of $(G, \mathbf{c})$ by 2-stable sets and join all elements of A_u to all elements of B_v if and only if there is a color class that consists of u and v. If $|A| > |B|$, we expand B with $|A| - |B|$ additional nodes, each joined to all nodes of A. Note that the way H is constructed guarantees that $|N(T)| \geq |T|$ for each $T \subseteq A$. Thus, by Theorem 2.1, H has a matching of size A. Also, at most $2t$ nodes of B are not contained in this matching. This is because, by Lemma 4.1, there are at most weight $2t$ color classes S of the fractional coloring that do not contain an element of K and $S \cap (V(G) \backslash P) \leq 1$. Now in an obvious manner, this matching and the node not saturated by it, if it exists, correspond to a coloring of $(G'', \mathbf{c}'')$ with at most $c(K) + 2t$ 2-stable sets. Thus, $\chi_2(G'', \mathbf{c}'') \leq c(K) + 2t$.

The proof is now complete:

$$\begin{aligned} \chi_2(G, \mathbf{c}) &\leq \chi_2(G'', \mathbf{c}'') + \chi_2(G', \mathbf{c}') \leq c(K) + 2t + \tfrac{1}{\beta}(\mathbf{a} \cdot \mathbf{c}) - c(K) + \tfrac{\eta(G')}{2} \\ &\leq \eta_2(G, \mathbf{c}) + \tfrac{\eta(G)-1}{2} + t \leq \eta_2(G, \mathbf{c}) + \tfrac{\eta(G)}{2}, \end{aligned}$$

since $\eta(G') \leq \eta(G) - 1$, $\frac{1}{\beta}(\mathbf{a} \cdot \mathbf{c}) + t = \eta_2(G, \mathbf{c})$, and $t \leq \frac{1}{2}$. ◻

Note that in the above proof, if $c(V(G)) < 2\eta_2(G, \mathbf{c})$, then there must be an inequality that is not of type (3) for which $\frac{1}{\beta}(\mathbf{a} \cdot \mathbf{c}) = \eta_2(G, \mathbf{c})$. Thus we are in the second case and $t = 0$. In this case the proof has demonstrated the following, slightly stronger, statement. We will use it in the next theorem.

COROLLARY 4.3. *Let G be a graph, $\mathbf{c} \in Z^{V(G)}$, $\mathbf{c} > \mathbf{0}$, and $r \geq \eta_2(G, \mathbf{c})$. If $c(V(G)) < 2r$, then $\chi_2(G, \mathbf{c}) < r + \frac{\eta(G)}{2}$.*

The upper bound of Theorem 4.2 is tight. If, for instance, G is the complete p-partite graph with each color class having an odd number of nodes, then $\chi_2(G) = \eta_2(G) + \frac{\eta(G)}{2}$.

We note that for any graph G and any vector $\mathbf{c} \in Z_+^{V(G)}$, $\chi_2(G, \mathbf{c})$ can be computed efficiently using the theory of matchings. First observe that any $v \in V(G)$ such that $N(v) \cup \{v\} = V(G)$ will account for $c(v)$ color classes in any optimal coloring of $(G, \mathbf{c})$ by 2-stable sets. Thus, we can ignore them and assume that every node of G belongs to a stable set of size 2. As a consequence, we may further assume that in any optimal coloring of $(G, \mathbf{c})$ by 2-stable sets, all color classes are of cardinality 2. Thus optimal colorings of $(G, \mathbf{c})$ correspond to optimal edge coverings of $(\bar{G}, \mathbf{c})$. Now in turn, it is well known that optimal edge coverings of $(\bar{G}, \mathbf{c})$ can be computed with the aid of optimal b-matchings of $(G, \mathbf{c})$. Namely, given an optimal b-matching $\mathbf{x}$ of $(\bar{G}, \mathbf{c})$, greedily construct a vector $\mathbf{x}' \in Z_+^{E(\bar{G})}$ such that $\mathbf{x} + \mathbf{x}'$ is an edge covering of $(\bar{G}, \mathbf{c})$. (Note that $\mathbf{1} \cdot (\mathbf{x} + \mathbf{x}') = \mathbf{1} \cdot \mathbf{c}$.) It is straightforward to verify that $\mathbf{x} + \mathbf{x}'$ is an optimal edge cover of $(\bar{G}, \mathbf{c})$. We note finally that the problem of finding an optimal b-matching in $(\bar{G}, \mathbf{c})$ can be solved efficiently using Edmonds's matching polyhedron theorem [6].

We would like to mention that the relationship between b-matchings and edge coverings outlined here is based on Gallai's theorem (see [3]) concerning the case where $\mathbf{c}$ is the all-ones vector. Also, this relationship can be extended to the fractional counterparts of edge coverings and b-matchings. In this case, for any graph G and any positive $\mathbf{c} \in Z^{V(G)}$ the inequality of Theorem 4.2 implies that

$$\mu(G,\mathbf{c}) \geq \mu'(G,\mathbf{c}) - \tfrac{\eta(H)}{2},$$

where $\mu(G,\mathbf{c})$ and $\mu'(G,\mathbf{c})$ are the values of an optimal b-matching and an optimal fractional b-matching of $(G,\mathbf{c})$, respectively, and $H = \bar{G} - \{v : N(v) = \emptyset\}$.

We now turn our attention to colorings by 3-stable sets.

THEOREM 4.4. *For any bipartite graph G and $\mathbf{c} \in Z_+^{V(G)}$, $\chi_3(G,\mathbf{c}) \leq \lceil \eta_3(G,\mathbf{c}) \rceil + 1$.*

The crux of the proof is the following result.

LEMMA 4.5. *Let G be a bipartite graph and $\mathbf{c} \in Z_+^{V(G)}$. If one of the following two conditions holds, then $\chi_3(G,\mathbf{c}) = \lceil \eta_3(G,\mathbf{c}) \rceil$.*

(i) *$c(V(G)) < \eta_3(G,\mathbf{c})$ and $\frac{1}{\beta}(\mathbf{a} \cdot \mathbf{c}) = \eta_3(G,\mathbf{c})$ for an inequality $\mathbf{a} \cdot \mathbf{x} \leq \beta$ of $\mathbf{A}_3\mathbf{x} \leq \mathbf{b}$, which is not of type* (1).

(ii) *G can be obtained from a 2-star by deleting zero or more nodes.*

Proof. Any coloring is also a fractional coloring and thus $\chi_3(G,\mathbf{c}) \geq \lceil \eta_3(G,\mathbf{c}) \rceil$. We show the reverse inequality by induction on $\eta_3(G,\mathbf{c})$. That is, we assume that the lemma holds for any $\mathbf{c}' \in Z_+^{V(G)}$ such that $\eta_3(G,\mathbf{c}') \leq \lceil \eta_3(G,\mathbf{c}) \rceil - 1$. With no loss of generality, $c(v) > 0$ for all $v \in V(G)$. Denote by $\mathcal{S}$ the color classes of an optimal fractional coloring $\mathbf{y}$ of $(G,\mathbf{c})$. We distinguish two cases.

Case 1. $(G,\mathbf{c})$ fulfills condition (ii).

Suppose that G is a 2-star. With the notation of Definition 3.4, we assume that $|R| \geq |L|$. Let S be a member of $\mathcal{S}$ that contains v_1. If necessary, include additional nodes in S so that it is either maximal or $|S| = 3$. Let $\mathbf{a} \cdot \mathbf{x} \leq \beta$ be an inequality of $\mathbf{A}_3\mathbf{x} \leq \mathbf{b}$. If $\mathbf{a} \cdot \mathbf{c} = \beta\eta_3(G,\mathbf{c})$, then by Lemma 4.1, $\mathbf{a} \cdot \mathbf{c} - \mathbf{a} \cdot \chi^S = \beta(\eta_3(G,\mathbf{c}) - 1)$. Otherwise, it can be readily checked that $\mathbf{a} \cdot \mathbf{c} - \mathbf{a} \cdot \chi^S \leq \beta(\lceil \eta_3(G,\mathbf{c}) \rceil - 1)$. Thus $\eta_3(G,\mathbf{c} - \chi^S) \leq \lceil \eta_3(G,\mathbf{c}) \rceil - 1$ and, by induction, $\chi_3(G,\mathbf{c} - \chi^S) \leq \lceil \eta_3(G,\mathbf{c} - \chi^S) \rceil$. It follows that

$$\chi_3(G,\mathbf{c}) \leq \chi_3(G,\mathbf{c} - \chi^S) + 1 \leq \lceil \eta_3(G,\mathbf{c} - \chi^S) \rceil + 1 \leq \lceil (\eta_3(G,\mathbf{c}) \rceil,$$

as required.

So assume that G is not a 2-star. If there is a node v such that $|N(v)| \geq 4$, let S be a member of $\mathcal{S}$ that contains v. Otherwise, let S be any member of $\mathcal{S}$. It is routine to check that $\chi_3(G,\mathbf{c} - \chi^S) \leq \lceil \eta_3(G,\mathbf{c}) \rceil - 1$. Again the theorem follows by induction.

Case 2. G does not fulfill condition (ii).

Let $\mathbf{a} \cdot \mathbf{x} \leq \beta$ be such that $\frac{1}{\beta}(\mathbf{a} \cdot \mathbf{c}) = \eta_3(G,\mathbf{c})$. Denote by P the nodes of G that do not belong to the support of $\mathbf{a}$. Our goal will be to reduce $(G,\mathbf{c})$ into two weighted graphs $(G',\mathbf{c}')$ and $(G'',\mathbf{c}'')$ and then color the first one using induction and the second one using Corollary 4.3.

$$\text{Let } \mathcal{S}' = \begin{cases} \{S \in \mathcal{S} : S \cap P \neq \emptyset \text{ or } |S| = 2\} & \text{if } \mathbf{a} \cdot \mathbf{x} \leq \beta \text{ is of type (3b)}, \\ \{S \in \mathcal{S} : S \cap P \neq \emptyset\} & \text{otherwise.} \end{cases}$$

Let $p = \sum_{S \in \mathcal{S}'} y_S$. By Lemma 4.1 and by the definition of $\mathbf{a} \cdot \mathbf{x} \leq \beta$, there exists a node $v \in V(G)$ such that $v \in S$ for all $S \in \mathcal{S}'$. (When $\mathbf{a} \cdot \mathbf{x} \leq \beta$ is of type (3a) or (3b) v is a node with coefficient 2 or 3, respectively, and in the other cases it is any node from the set T of the support of $\mathbf{a}$.) Thus v is not adjacent to any other node that belongs to a member of $\mathcal{S}'$. Let G' be the graph induced by the nodes different from

v that belong to some member of $\mathcal{S}'$, and let $\mathbf{c}'$ be the restriction of $\sum_{S\in\mathcal{S}'} y_S\chi^S$ to $V(G')$. Let G'' be the graph induced by the support of $\mathbf{a}$, and let $\mathbf{c}''$ be the restriction of $\mathbf{c} - \sum_{S\in\mathcal{S}'} y_S\chi^S - \lceil p\rceil\chi^{\{v\}}$ to $V(G'')$.

Suppose for the moment that both $\mathbf{c}'$ and $\mathbf{c}''$ are integral (so $p = \lceil p\rceil$). Because $c(V(G)) \leq 3\eta_3(G,\mathbf{c})$, not every $S \in \mathcal{S}'$ is of size 3. Thus $c'(V(G')) < 2p$ and by Corollary 4.3, $(G,\mathbf{c}')$ has a coloring with at most p 2-stable sets. Each one of these stable sets can be extended to include v. In addition, $\mathbf{y}$ restricted to $\mathcal{S}\backslash\mathcal{S}'$ gives a coloring of $(G'',\mathbf{c}'')$ of value $\eta_3(G,\mathbf{c}) - p$. Moreover, $(G'',\mathbf{c}'')$ fulfills condition (ii); therefore, by Case 1, $(G'',\mathbf{c}'')$ has a coloring of value $\lceil\eta_3(G,\mathbf{c}) - p\rceil$. Combining these two colorings gives a coloring of $(G,\mathbf{c})$ of value $\lceil\eta_3(G,\mathbf{c}) - p\rceil + p = \lceil\eta_3(G,\mathbf{c})\rceil$, as required.

To conclude the case and the lemma, we now show that, if necessary, the coloring vector $\mathbf{y}$ can be altered (while keeping $\mathcal{S}$ unchanged) so that it is still an optimal fractional coloring and both $\mathbf{c}'$ and $\mathbf{c}''$ are integral.

If $\mathbf{c}'$ and $\mathbf{c}''$ are nonintegral, then $\mathbf{a}\cdot\mathbf{x} \leq \beta$ must be an inequality of the form $2x(v) + x(N(v)) + x(K) \leq 3$, where v is the special node identified earlier and K has exactly two elements u and w. This follows from Lemma 4.1 and the fact that $c(P)$ is integral. (The mentioned inequality is the only one for which there can be more than one color class of size less than 3 in the optimal fractional coloring of its support.) Note that since $\mathbf{c}$ is integral, u and w index the only nonintegral components of $\mathbf{c}'$ and $\mathbf{c}''$. Let $p_u = \sum_{\{S\in\mathcal{S}':u\in S\}} y_S$ and $p_w = \sum_{\{S\in\mathcal{S}':w\in S\}} y_S$. By Lemma 4.1, any member of $\mathcal{S}$ that contains v must also contain u or w, so $p_u + p_w = c(v)$ and $p = p_u + p_w$. So p is an integer and, by assumption, p_u and p_w are both not integers. Consider the set of nodes $Q \subseteq P$ that belong to some member of $\mathcal{S}'$ together with $\{v,u\}$. Since $c(Q)$ is an integer, either $y_{\{u,v\}} \notin \mathrm{Z}$ or there exists a $z \in Q$ such that $y_{\{u,v,z\}} \notin \mathrm{Z}$. In the former case, if $y_{\{u,v\}} \notin \mathrm{Z}$, then since $c(P) = \sum_{\{S\in\mathcal{S}':|S|=3\}} y_S \in \mathrm{Z}$ we have that $y_{\{v,u\}} + y_{\{v,w\}} \in \mathrm{Z}$. Now decrease $y_{\{v,u\}}$ by $y_{\{v,u\}} - \lfloor y_{\{v,u\}}\rfloor$ and increase $y_{\{v,w\}}$ by the same amount. In the latter case, $\{v,w,z\}$ must also be in $\mathcal{S}'$ since $c(z) \in \mathrm{Z}$ and $y_{\{v,u,z\}} \notin \mathrm{Z}$. Then decrease $y_{\{v,u,z\}}$ by $y_{\{v,u,z\}} - \lfloor y_{\{v,u,z\}}\rfloor$ and increase $y_{\{v,w,z\}}$ by the same amount. Repetition of this procedure yields the desired coloring. □

Proof of Theorem 4.4. We prove the theorem by induction on $\lceil\eta_3(G,\mathbf{c})\rceil$. If $(G,\mathbf{c})$ fulfills one of the conditions of Lemma 4.5, we are done. If not, then choose any color class S from an optimal fractional coloring of $(G,\mathbf{c})$ by 3-stable sets and, if necessary, add nodes until S is either maximal of size 3. If $\eta_3(G,\mathbf{c}-\chi^S) < \lceil\eta_3(G,\mathbf{c})\rceil - 1$, then we can apply induction and find a coloring of $(G,\mathbf{c}-\chi^S)$ with $\lceil\eta_3(G,\mathbf{c}-\chi^S)\rceil + 1$ 3-stable sets. Otherwise, it must be that $(G,\mathbf{c}-\chi^S)$ fulfills condition (i) of Lemma 4.5 and the result follows. □

Again, the upper bound of the theorem is tight, for if G is a complete bipartite graph where every color class is larger than 3 and equal to 1 mod 3, then $\chi_3(G) = \lceil\eta_3(G)\rceil + 1$.

Acknowledgments. We thank P. Hansen for introducing us to the subject and O. Marcotte for helpful discussions.

REFERENCES

[1] F. Barahona, J. Fonlupt, and A.R. Mahjoub, *Compositions of graphs and polyhedra* IV*: Acyclic spanning subgraphs,* SIAM J. Discrete Math., 7 (1994), pp. 390–402.

[2] S. Baum and L.E. Trotter Jr., *Integer rounding for polymatroid and branching optimization problems*, SIAM J. Alg. Discrete Methods, 2 (1981), pp. 416–425.

[3] J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications*, North–Holland, New York, 1976.
[4] B.-L. Chen and K.-W. Lih, *A note on the m-bounded chromatic number of a tree,* European J. Combin., 14 (1993), pp. 311–312.
[5] W. Cook, private communication.
[6] J. Edmonds, *Maximum matching and a polyhedron with* $0,1$ *vertices,* J. Res. Nat. Bureau Standards, 69 (1965), pp. 125–130.
[7] P. Hansen, A. Hertz, and J. Kuplinsky, *Bounded vertex colorings of graphs,* Discrete Math., 111 (1993), pp. 305–312.
[8] T.R. Jensen and B. Toft, *Graph Coloring Problems*, John Wiley, Toronto, 1995.
[9] K. Kilakos, *Colorings and Fractional Colorings,* Ph.D. thesis, University of Waterloo, Waterloo, Ontario, Canada, 1993.
[10] K. Kilakos and O. Marcotte, *Fractional and integral colorings,* Math. Programming Ser. A, 76 (1997), pp. 333–347.
[11] J. Krarup and D. de Werra, *Chromatic optimization: Limitations, objectives, uses, references,* European J. Oper. Res., 11 (1982), pp. 1–9.
[12] L. Lovász, *Kneser's conjecture, chromatic number and homotopy,* J. Combin. Theory Ser. B, 25 (1978), pp. 319–324.
[13] M.W. Padberg, *On the facial structure of the set packing polydedra,* Math. Programming, 5 (1973), pp. 199–215.
[14] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley, Chichester, UK, 1986.
[15] F.B. Shepherd, *Near-Perfection and Stable Set Polyhedra*, Ph.D. thesis, University of Waterloo, Waterloo, Ontario, Canada, 1990.
[16] F.B. Shepherd, *Near-perfect matrices,* Math. Programming, 64 (1994), pp. 295–323.

SIAM J. Discrete Math.
Vol. 12, No. 2, pp. 276–287

INDEPENDENT SETS IN ASTEROIDAL TRIPLE-FREE GRAPHS*

HAJO BROERSMA†, TON KLOKS‡, DIETER KRATSCH§, AND HAIKO MÜLLER§

Abstract. An asteroidal triple (AT) is a set of three vertices such that there is a path between any pair of them avoiding the closed neighborhood of the third. A graph is called AT-free if it does not have an AT. We show that there is an $O(n^4)$ time algorithm to compute the maximum weight of an independent set for AT-free graphs. Furthermore, we obtain $O(n^4)$ time algorithms to solve the INDEPENDENT DOMINATING SET and the INDEPENDENT PERFECT DOMINATING SET problems on AT-free graphs. We also show how to adapt these algorithms such that they solve the corresponding problem for graphs with bounded asteroidal number in polynomial time. Finally, we observe that the problems CLIQUE and PARTITION INTO CLIQUES remain NP-complete when restricted to AT-free graphs.

Key words. graph algorithms, AT-free graphs, independent set, independent dominating set

AMS subject classifications. 68R10, 05C85

PII. S0895480197326346

1. Introduction.

Asteroidal triples (ATs) were introduced in 1962 to characterize interval graphs as those chordal graphs that do not contain an AT [20]. Graphs not containing an AT are called AT-free graphs. They form a large class of graphs containing interval, permutation, trapezoid, and cocomparability graphs. Since 1989, AT-free graphs have been studied extensively by Corneil, Olariu, and Stewart. They have published a collection of papers presenting many structural and algorithmic properties of AT-free graphs (see, e.g., [6, 7]). Further results on AT-free graphs were obtained in [2, 18, 23].

Until now, knowledge of the algorithmic complexity of NP-complete graph problems when restricted to AT-free graphs was relatively small compared to that of other graph classes. The problems TREEWIDTH, PATHWIDTH, and MINIMUM FILL-IN remain NP-complete on AT-free graphs [1, 28]. On the other hand, various domination-type problems like CONNECTED DOMINATING SET [2, 7], CARDINALITY STEINER TREE [2], DOMINATING SET [19], and TOTAL DOMINATING SET [19] can be solved by polynomial time algorithms for AT-free graphs. However, there is a collection of classical NP-complete graph problems for which the algorithmic complexity when restricted to AT-free graphs was not known. Prominent representatives are INDEPENDENT SET, CLIQUE, GRAPH k-COLORABILITY, PARTITION INTO CLIQUES, HAMILTONIAN CIRCUIT, and HAMILTONIAN PATH.

A crucial reason for the lack of progress in designing efficient algorithms for NP-complete problems on AT-free graphs seemed to be that none of the typical representations, which are useful for the design of efficient algorithms on special graph classes, are known for AT-free graphs. Contrary to well-known graph classes such as chordal, permutation, and circular-arc graphs, no geometric representation of AT-free

*Received by the editors August 20, 1997; accepted for publication (in revised form) August 11, 1998; published electronically April 29, 1999.
http://www.siam.org/journals/sidma/12-2/32634.html
†University of Twente, Faculty of Applied Mathematics, P.O. Box 217, 7500 AE Enschede, the Netherlands (H.J.Broersma@math.utwente.nl).
‡Department of Applied Mathematics and DIMATIA, Charles University, Malostranské nám. 25, 11800 Praha 1, Czech Republic (ton@kam.ms.mff.cuni.cz).
§Friedrich-Schiller-Universität Jena, Fakultät für Mathematik und Informatik, 07740 Jena, Germany (kratsch@minet.uni-jena.de, hm@minet.uni-jena.de).

graphs is known. Furthermore no representation of AT-free graphs by an elimination scheme of vertices or edges, small separators, or a small number of certain separators is known. Fortunately, it turns out that the design of all our algorithms is supported by a simple structural property of AT-free graphs that can be obtained from the definition of AT-free graphs rather easily.

Our approach in this paper is similar to the one used to design algorithms for problems such as TREEWIDTH [14, 17], MINIMUM FILL-IN [17], and VERTEX RANKING [18] on AT-free graphs. However, these algorithms have polynomial running time only under the additional constraint that the number of minimal separators is bounded by a polynomial in the number of vertices of the graph. (Notice that all three problems are NP-complete on AT-free graphs.) Technically, for the three different independent set problems in this paper, we are able to replace the set of all minimal separators, used in [14, 17, 18] (which might be "too large" in size) by the "small" set of all closed neighborhoods of the vertices of the graph.

Finding out the algorithmic complexity of INDEPENDENT SET on AT-free graphs is a challenging task. Besides the fact that INDEPENDENT SET is a classical and well-studied NP-complete problem, the problem is also interesting because, contrary to well-known subclasses of AT-free graphs such as cocomparability graphs, not all AT-free graphs are perfect. Thus the polynomial time algorithm for perfect graphs of Grötschel, Lovász, and Schrijver [11] that solves the INDEPENDENT SET problem does not apply to AT-free graphs.

We present the first polynomial time algorithm solving the NP-complete problem INDEPENDENT SET, when restricted to AT-free graphs. More precisely, our main result is an $O(n^4)$ algorithm to compute the maximum weight of an independent set in an AT-free graph. Furthermore, we present $O(n^4)$ time algorithms to solve the problem INDEPENDENT DOMINATING SET and INDEPENDENT PERFECT DOMINATING SET (also called EFFICIENT DOMINATING SET). We also observe that the problems CLIQUE and PARTITION INTO CLIQUES remain NP-complete when restricted to AT-free graphs.

A natural generalization of ATs are the so-called asteroidal sets. Structural results for asteroidal sets and algorithms for graphs with bounded asteroidal number were obtained in [15, 21, 25, 27]. Computing the asteroidal number (i.e., the maximum cardinality of an asteroidal set) turns out to be NP-complete in general, but solvable in polynomial time for many graph classes [16]. Furthermore, the results for problems such as TREEWIDTH and MINIMUM FILL-IN on AT-free graphs can be generalized to graphs with bounded asteroidal number [15].

We show how to adapt our algorithms to obtain polynomial time algorithms for graphs with bounded asteroidal number solving the problems INDEPENDENT SET, INDEPENDENT DOMINATING SET, and INDEPENDENT PERFECT DOMINATING SET.

2. Preliminaries. We denote the number of vertices of a graph $G = (V, E)$ by n and the number of edges by m.

Recall that an independent set in a graph G is a set of pairwise nonadjacent vertices. The independence number of a graph G denoted by $\alpha(G)$ is the maximum cardinality of an independent set in G.

For a graph $G = (V, E)$ and $W \subseteq V$, $G[W]$ denotes the subgraph of G induced by the vertices of W; we write $\alpha(W)$ for $\alpha(G[W])$. For convenience, for a vertex x of G we write $G - x$ instead of $G[V \setminus \{x\}]$. Analogously, for a subset $X \subseteq V$, we write $G - X$ instead of $G[V \setminus X]$. We consider components of a graph as maximal connected subgraphs as well as vertex subsets. For a vertex x of $G = (V, E)$, $N(x) = \{y \in V : \{x, y\} \in E\}$ is the neighborhood of x and $N[x] = N(x) \cup \{x\}$ is the closed neighborhood of x. For $W \subseteq V$, $N[W] = \bigcup_{x \in W} N[x]$.

A set $S \subseteq V$ is a separator of the graph $G = (V, E)$ if $G - S$ is disconnected.

DEFINITION 2.1. *Let $G = (V, E)$ be a graph. A set $\Omega \subseteq V$ is an* asteroidal set *if for every $x \in \Omega$ the set $\Omega \setminus \{x\}$ is contained in one component of $G - N[x]$. An asteroidal set with three vertices is called an* AT.

Notice that every asteroidal set is an independent set.

Remark 2.1. A triple $\{x, y, z\}$ of vertices of G is an AT if and only if for every two of these vertices there is a path between them avoiding the closed neighborhood of the third.

DEFINITION 2.2. *A graph $G = (V, E)$ is called* AT-free *if G has no AT.*

It is well known that the INDEPENDENT SET problem, "Given a graph G and a positive integer k, decide whether $\alpha(G) \geq k$," is NP-complete [9]. The problem remains NP-complete, even when restricted to triangle-free, 3-connected, cubic planar graphs [26]. Moreover, the independence number is hard to approximate within a factor of $n^{1-\epsilon}$ for any constant $\epsilon > 0$ [12]. Despite this discouraging recent result on the complexity of approximation, the independence number can be computed in polynomial time on many special classes of graphs (see [13]). For example, the best known algorithm to solve the problem on cocomparability graphs has running time $O(n + m)$ (see [24]).

The main result of this paper is an $O(n^4)$ algorithm to compute the maximum weight of an independent set in an AT-free graph with real vertex weights. The structural properties enabling the design of our algorithms are given in the next three sections. For convenience, we deal with the cardinality case of our problems first, and point out how to extend the method to graphs with real vertex weights in section 9.

3. Intervals. Let $G = (V, E)$ be an AT-free graph, and let x and y be two distinct nonadjacent vertices of G. Throughout the paper we use $C^x(y)$ to denote the component of $G - N[x]$ containing y, and $r(x)$ to denote the number of components of $G - N[x]$.

DEFINITION 3.1. *A vertex $z \in V \setminus \{x, y\}$ is* between x *and y if x and z are in one component of $G - N[y]$ and y and z are in one component of $G - N[x]$.*

Equivalently, z is between x and y in G if there is an x, z-path avoiding $N[y]$ and there is an y, z-path avoiding $N[x]$.

DEFINITION 3.2. *The* interval $I = I(x, y)$ *of G is the set of all vertices of G that are between x and y.*

Thus $I(x, y) = C^x(y) \cap C^y(x)$.

4. Splitting intervals. Let $G = (V, E)$ be an AT-free graph; let $I = I(x, y)$ be a nonempty interval of G; and let $s \in I$. Let $I_1 = I(x, s)$ and $I_2 = I(s, y)$.

LEMMA 4.1. *x and y are in different components of $G - N[s]$.*

Proof. Assume x and y would be in the same component of $G - N[s]$. Then there is an x, y-path avoiding $N[s]$. However, $s \in I$ implies that there is an s, y-path avoiding $N[x]$ and an s, x-path avoiding $N[y]$. Thus $\{s, x, y\}$ is an AT of G, which is a contradiction. ∎

COROLLARY 4.2. *$I_1 \cap I_2 = \emptyset$.*

Proof. Assume $z \in I_1 \cap I_2$. Then $z \in I_1$ implies that there is a component C^s of $G - N[s]$ containing both x and z. Furthermore, $z \in I_2$ implies also that $y \in C^s$, contradicting Lemma 4.1. ∎

LEMMA 4.3. *$I_1 \subseteq I$ and $I_2 \subseteq I$.*

Proof. Let $z \in I_1$. Clearly $s \in I$ implies $s \in C^x(y)$. Thus $z \in I_1$ implies $z \in C^x(y)$. Clearly $z \in C^s(x)$ since $z \in I_1$. By Lemma 4.1, $C^s(x)$ is contained in a component of $G - N[y]$, and obviously this component contains x. This proves $z \in I$. Consequently $I_1 \subseteq I$.

$I_2 \subseteq I$ can be shown analogously. ◻

THEOREM 4.4. *There exist components $C_1^s, C_2^s, \ldots, C_t^s$ of $G - N[s]$ such that*

$$I \setminus N[s] = I_1 \cup I_2 \cup \bigcup_{i=1}^{t} C_i^s.$$

Proof. By Lemma 4.3, we have $I_1 \subseteq I \setminus N[s]$ and $I_2 \subseteq I \setminus N[s]$. By Lemma 4.1, x and y belong to different components $C^s(x)$ and $C^s(y)$ of $G - N[s]$. Let $z \in I \setminus N[s]$.

Assume $z \in C^s(x)$. There is a z, y-path avoiding $N[x]$. This path must contain a vertex of $N[s]$, showing the existence of a z, s-path avoiding $N[x]$. Hence $z \in I_1$.

Similarly $z \in C^s(y)$ implies $z \in I_2$.

Assume $z \notin C^s(x)$ and $z \notin C^s(y)$. Since $z \notin N[s]$, z belongs to the component $C^s(z)$ of $G - N[s]$. For any vertex $p \in C^s(z)$, there is a p, z-path avoiding $N[x]$, since $C^s(z) \neq C^s(x)$. Since $z \in I$, there is a z, y-path avoiding $N[x]$. Hence there is also a p, y-path avoiding $N[x]$. This shows $C^s(z) \subseteq I \setminus N[s]$. ◻

COROLLARY 4.5. *Every component of $G[I \setminus (N[s] \cup I_1 \cup I_2)]$ is a component of $G - N[s]$.*

5. Splitting components. Let $G = (V, E)$ be an AT-free graph. Let C^x be a component of $G - N[x]$ and let y be a vertex of C^x. Thus $C^x = C^x(y)$. We study the components of the graph $C^x - N[y]$.

THEOREM 5.1. *Let D be a component of the graph $C^x - N[y]$. Then $N[D] \cap (N[x] \setminus N[y]) = \emptyset$ if and only if D is a component of $G - N[y]$.*

Proof. Let D be a component of $C^x - N[y]$ with $N[D] \cap (N[x] \setminus N[y]) = \emptyset$. Since no vertex of D has a neighbor in $N[x] \setminus N[y]$, D is a component of $G - N[y]$.

Now let $D \subseteq C^x$ be a component of $G - N[y]$. Then $N[D] \cap N[x] \subseteq N[y]$. ◻

COROLLARY 5.2. *Let B be a component of the graph $C^x - N[y]$. Then $N[B] \cap (N[x] \setminus N[y]) \neq \emptyset$ if and only if $B \subseteq C^y(x)$.*

THEOREM 5.3. *Let $B_1, \ldots, B_\ell$ denote the components of the graph $C^x - N[y]$ that are contained in $C^y(x)$. Then $I(x, y) = \bigcup_{i=1}^{\ell} B_i$.*

Proof. Let $I = I(x, y)$. First we show that $B_i \subseteq I$ for every $i \in \{1, \ldots, \ell\}$. Let $z \in B_i$. There is an x, z-path avoiding $N[y]$, since some vertex in B_i has a neighbor in $N[x] \setminus N[y]$. Clearly, there is also a z, y-path avoiding $N[x]$, since z and y are both in C^x. This shows that $z \in I$. Consequently $\bigcup_{i=1}^{\ell} B_i \subseteq I$.

Suppose $z \in I \setminus \bigcup_{i=1}^{\ell} B_i$. Since $z \notin \bigcup_{i=1}^{\ell} B_i$, the component D of $C^x - N[y]$ that contains z does not contain a vertex with a neighbor in $N[x] \setminus N[y]$. Thus $z \notin C^y(x)$, implying $z \notin I$, a contradiction. ◻

6. Computing the independence number. In this section we describe our algorithm to compute the independence number of an AT-free graph. The algorithm we propose uses dynamic programming on intervals and components. All intervals and all components are sorted according to a nondecreasing number of vertices. Following this order, the algorithm determines the independence number of each component and of each interval using the formulas given in Lemmas 6.1, 6.2, and 6.3.

We start with an obvious lemma.

LEMMA 6.1. *Let $G = (V, E)$ be any graph. Then*

$$\alpha(G) = 1 + \max_{x \in V} \left(\sum_{i=1}^{r(x)} \alpha(C_i^x) \right),$$

where $C_1^x, C_2^x, \ldots, C_{r(x)}^x$ are the components of $G - N[x]$.

Applying Lemma 6.1 to the decomposition given by Theorems 5.1 and 5.3, we obtain the following lemma.

LEMMA 6.2. *Let $G = (V, E)$ be an AT-free graph. Let $x \in V$ and let C^x be a component of $G - N[x]$. Then*

$$\alpha(C^x) = 1 + \max_{y \in C^x} \left(\alpha(I(x, y)) + \sum_i \alpha(D_i^y) \right),$$

where the D_i^y's are the components of $G - N[y]$ contained in C^x.

Applying Lemma 6.1 to the decomposition given by Theorem 4.4, we obtain the following lemma.

LEMMA 6.3. *Let $G = (V, E)$ be an AT-free graph. Let $I = I(x, y)$ be an interval of G. If $I = \emptyset$ then $\alpha(I) = 0$. Otherwise*

$$\alpha(I) = 1 + \max_{s \in I} \left(\alpha(I(x, s)) + \alpha(I(s, y)) + \sum_i \alpha(C_i^s) \right),$$

where the C_i^s's are the components of $G - N[s]$ contained in $I(x, y)$.

Remark 6.1. Notice that the components D_i^y and C_i^s as well as the intervals $I(x, s)$ and $I(s, y)$ on the right-hand sides of the formulas in Lemmas 6.2 and 6.3 are proper subsets of C^x and I, respectively. Hence $\alpha(C^x)$ (respectively, $\alpha(I)$) can be computed by table look-up to components and intervals with a smaller number of vertices.

Consequently we obtain the following algorithm to compute the independence number $\alpha(G)$ for a given AT-free graph $G = (V, E)$, which is based on dynamic programming.

Step 1. For every $x \in V$ compute all components $C_1^x, C_2^x, \ldots, C_{r(x)}^x$ of $G - N[x]$.

Step 2. For every pair of nonadjacent vertices x and y compute the interval $I(x, y)$.

Step 3. Sort all the components and intervals according to nondecreasing number of vertices.

Step 4. Compute $\alpha(C)$ and $\alpha(I)$ for each component C and each interval I in the order of Step 3.

Step 5. Compute $\alpha(G)$.

THEOREM 6.4. *There is an $O(n^4)$ time algorithm to compute the independence number of a given AT-free graph.*

Proof. The correctness of the algorithm follows from the formulas of Lemmas 6.1, 6.2, and 6.3 as well as the order of the dynamic programming.

We show how to obtain the stated time complexity. Clearly, Step 1 can be implemented such that it takes $O(n(n+m))$ time using a linear time algorithm to compute the components of the graph $G - N[x]$ for each vertex x of G. For each component of $G - N[x]$, a sorted linked list of all its vertices and its number of vertices is stored. For all nonadjacent vertices x and y there is a pointer $P(x, y)$ to the list of $C^x(y)$. Thus in Step 2, an interval $I(x, y)$ can be computed using the fact that

$I(x,y) = C^x(y) \cap C^y(x)$. Hence a sorted vertex list of $I(x,y)$ can be computed in time $O(n)$ for each interval. Consequently the overall time bound for Step 2 is $O(n^3)$. There are at most n^2 components and at most n^2 intervals and each has at most n vertices. Thus, using the linear time sorting algorithm bucket sort, Step 3 can be done in time $O(n^2)$.

The bottleneck for the time complexity of our algorithm is Step 4. First consider a component C^x of $G - N[x]$ and a vertex $y \in C^x$. We need to compute the components of $G - N[y]$ that are contained in C^x. Each component D of $G - N[y]$ except $C^y(x)$ is contained in C^x if and only if $D \cap C^x \neq \emptyset$. Thus the components D of $G - N[y]$ with $D \subseteq C^x$ are exactly those components of $G - N[y]$ addressed by $P(y,z)$ for some $z \in C^x$. Thus all such components can be found in time $O(|C^x|)$ for fixed vertices x and $y \in C^x$. Hence the computation of $\alpha(C)$ for all components C takes time $\sum_{\{x,y\} \notin E} O(|C^x(y)|) = O(n^3)$.

Now consider an interval $I = I(x,y)$, and a vertex $s \in I$. We need to add up the independence numbers of the components C_i^s of $G - N[s]$ that are contained in I. The components of $G - N[y]$ that are contained in I are exactly those components addressed by $P(y,z)$ for some $z \in I$, except $C^s(x)$ and $C^s(y)$. Thus all such components can be found in time $O(|I(x,y)|)$ for a fixed interval $I(x,y)$ and $s \in I(x,y)$. Hence the computation of $\alpha(I)$ for all intervals I takes time $\sum_{\{x,y\} \notin E} \sum_{s \in I(x,y)} O(|I(x,y)|) = O(n^4)$.

Clearly Step 5 can be done in $O(n^2)$ time. Thus the running time of our algorithm is $O(n^4)$. ∎

7. Independent domination. The approach used to design the presented polynomial time algorithm to compute the independence number for AT-free graphs can also be used to obtain a polynomial time algorithm solving the INDEPENDENT DOMINATING SET problem on AT-free graphs. The best known algorithm to solve the weighted version of the problem on cocomparability graphs has running time $O(n^{2.376})$ [4].

DEFINITION 7.1. *Let $G = (V,E)$ be a graph. Then $S \subseteq V$ is a* dominating set *of G if every vertex of $V \setminus S$ has a neighbor in S. A dominating set $S \subseteq V$ is an* independent dominating set *of G if S is an independent set.*

We denote by $\gamma_{\mathrm{i}}(G)$ the minimum cardinality of an independent dominating set of the graph G. Given an AT-free graph G, our next algorithm computes $\gamma_{\mathrm{i}}(G)$. It works very similarly to the algorithm of the previous section.

We present only the formulas used in Steps 4 and 5 of the algorithm (which are similar to those in Lemmas 6.1, 6.2, and 6.3).

LEMMA 7.2. *Let $G = (V,E)$ be a graph. Then*

$$\gamma_{\mathrm{i}}(G) = 1 + \min_{x \in V} \left(\sum_{j=1}^{r(x)} \gamma_{\mathrm{i}}(C_j^x) \right),$$

where $C_1^x, C_2^x, \ldots, C_{r(x)}^x$ are the components of $G - N[x]$.

LEMMA 7.3. *Let $G = (V,E)$ be an AT-free graph. Let $x \in V$ and let C^x be a component of $G - N[x]$. Then*

$$\gamma_{\mathrm{i}}(C^x) = 1 + \min_{y \in C^x} \left(\gamma_{\mathrm{i}}(I(x,y)) + \sum_j \gamma_{\mathrm{i}}(D_j^y) \right),$$

where the D_j^y's are the components of $G - N[y]$ contained in C^x.

LEMMA 7.4. *Let $G = (V, E)$ be an AT-free graph. Let $I = I(x, y)$ be an interval. If $I = \emptyset$ then $\gamma_{\mathrm{i}}(I) = 0$. Otherwise*

$$\gamma_{\mathrm{i}}(I) = 1 + \min_{s \in I} \left(\gamma_{\mathrm{i}}(I(x,s)) + \gamma_{\mathrm{i}}(I(s,y)) + \sum_j \gamma_{\mathrm{i}}(C_j^s) \right),$$

where the C_j^s's are the components of $G - N[s]$ contained in $I(x, y)$.

The design and analysis of the algorithm is done similarly to the one in the previous section. This gives the following theorem.

THEOREM 7.5. *There exists an $O(n^4)$ time algorithm to compute the independence domination number γ_{i} of a given AT-free graph.*

8. Independent perfect domination. The INDEPENDENT PERFECT DOMINATING SET problem is a variant of the INDEPENDENT DOMINATING SET problem. The best known algorithm to solve the weighted version of the problem on cocomparability graphs has running time $O(n^2)$ [5].

DEFINITION 8.1. *A* perfect dominating set *of a graph $G = (V, E)$ is a set $S \subseteq V$ such that every vertex of $V \setminus S$ is adjacent to exactly one vertex in S. A perfect dominating set S is an* independent perfect dominating set *of G if S is an independent set. (An independent perfect dominating set is also called an* efficient dominating set.*)*

We denote the minimum cardinality of an independent perfect dominating set in G by $\gamma_{\mathrm{ip}}(G)$. If G does not have an independent perfect dominating set, we define $\gamma_{\mathrm{ip}}(G) = \infty$.

There is a close relationship between the problems INDEPENDENT PERFECT DOMINATING SET and INDEPENDENT DOMINATING SET which can often be exploited to transform an algorithm solving the INDEPENDENT DOMINATING SET problem into an algorithm solving the INDEPENDENT PERFECT DOMINATING SET problem. We demonstrate this for our algorithm of the previous section.

We present the formulas for an $O(n^4)$ algorithm to compute $\gamma_{\mathrm{ip}}(G)$ for a given AT-free graph G. Let x be a vertex of G and let C^x be a component of $G - N[x]$. Let $\Delta(x, C^x) = \{z \in C^x \mid d_G(z, x) > 2\}$. We denote by $\gamma_{\mathrm{ip}}(x, C^x)$ the minimum cardinality of an independent perfect dominating set S of C^x with $S \subseteq \Delta(x, C^x)$.

LEMMA 8.2. *Let $G = (V, E)$ be a graph. Then*

$$\gamma_{\mathrm{ip}}(G) = 1 + \min_{x \in V} \left(\sum_{i=1}^{r(x)} \gamma_{\mathrm{ip}}(x, C_i^x) \right),$$

where $C_1^x, C_2^x, \ldots, C_{r(x)}^x$ are the components of $G - N[x]$.

Let $I = I(x, y)$ be an interval of G. Let $\Delta(x, y, I) = \{z \in I \mid d_G(z, x) > 2 \wedge d_G(z, y) > 2\}$. We denote by $\gamma_{\mathrm{ip}}(x, y, I(x, y))$ the minimum cardinality of an independent perfect dominating set S of $G[I(x, y)]$ with $S \subseteq \Delta(x, y, I)$.

LEMMA 8.3. *Let $G = (V, E)$ be an AT-free graph. Let $x \in V$ and let C^x be a component of $G - N[x]$. If $\Delta(x, C^x) = \emptyset$ then $\gamma_{\mathrm{ip}}(x, C^x) = \infty$. If $\Delta(x, C^x) \neq \emptyset$ then*

$$\gamma_{\mathrm{ip}}(x, C^x) = 1 + \min_{y \in \Delta(x, C^x)} \left(\gamma_{\mathrm{ip}}(x, y, I(x, y)) + \sum_j \gamma_{\mathrm{ip}}(y, D_j^y) \right),$$

where the D_j^y's are the components of $G - N[y]$ contained in C^x.

LEMMA 8.4. *Let $G = (V, E)$ be an AT-free graph. Let $I(x, y)$ be an interval of G. For $s \in I$, let $I_1 = I(s, x)$ and $I_2 = I(s, y)$. If $\Delta(x, y, I) = \emptyset$ and $|I| = 0$ then $\gamma_{\text{ip}}(x, y, I) = 0$. If $\Delta(x, y, I) = \emptyset$ and $|I| > 0$ then $\gamma_{\text{ip}}(x, y, I) = \infty$. If $\Delta(x, y, I) \neq \emptyset$ then*

$$\gamma_{\text{ip}}(x, y, I) = 1 + \min_{s \in \Delta(x,y,I)} \left(\gamma_{\text{ip}}(x, s, I_1) + \gamma_{\text{ip}}(s, y, I_2) + \sum_j \gamma_{\text{ip}}(s, C_j^s) \right),$$

where the C_j^s's are the components of $G - N[s]$ contained in $I(x, y)$.

Our algorithm first computes the distance matrix of the given graph and then applies the approach of the previous two sections.

THEOREM 8.5. *There exists an $O(n^4)$ time algorithm to compute the independent perfect domination number γ_{ip} of a given AT-free graph.*

9. Weights on the vertices. In this section we consider AT-free graphs with real weights. Since we assume a unit-cost RAM as computational model, weights can be compared and added in constant time.

DEFINITION 9.1. *A* weighted *graph is a pair (G, w), where $G = (V, E)$ is a graph and every vertex x of G is assigned a real weight $w(x)$. Let $S \subseteq V$. Then $w(S) = \sum_{x \in S} w(x)$ is the weight of S.*

For a weighted graph (G, w) the maximum weight of an independent set of G is denoted by $\alpha^w(G)$, and the minimum weight of an independent dominating set of G is denoted by $\gamma_{\text{i}}^w(G)$. Clearly, $\alpha^w(G) = \alpha^w(G[\{x \in V\colon\ w(x) > 0\}])$.

First we prove a version of Lemma 6.1 extended to weighted graphs.

LEMMA 9.2. *Let (G, w) be a weighted graph, $G = (V, E)$. Then*

$$\alpha^w(G) = \max_{x \in V, w(x) > 0} \left(w(x) + \sum_{i=1}^{r(x)} \alpha^w(C_i^x) \right),$$

where $C_1^x, C_2^x, \ldots, C_{r(x)}^x$ are the components of $G - N[x]$.

Proof. If $w(x) \leq 0$ for all $x \in V$ then $\alpha^w(G) = 0$, since the empty set is independent. Otherwise G has a nonempty independent set S of maximum weight containing vertices of positive weight only. For such a set we have $x \in S$ if and only if $w(S) = w(x) + \sum_{i=1}^{r(x)} \alpha^w(C_i^x)$, where $C_1^x, C_2^x, \ldots, C_{r(x)}^x$ are the components of $G - N[x]$. ☐

The two remaining lemmas of section 6 generalize to weighted AT-free graphs in a similar way. We obtain the formulas

$$\alpha^w(C^x) = \max_{y \in C^x, w(y) > 0} \left(w(y) + \alpha^w(I(x, y)) + \sum_i \alpha^w(D_i^y) \right),$$

$$\alpha^w(I) = \max_{s \in I, w(s) > 0} \left(w(s) + \alpha^w(I(x, s)) + \alpha^w(I(s, y)) + \sum_i \alpha^w(C_i^s) \right),$$

analogously to the formulas in Lemmas 6.2 and 6.3, respectively. Therefore, the algorithm given in section 6 applied to a weighted AT-free graph computes the maximum weight of an independent set and runs in time $O(n^4)$.

For the problem INDEPENDENT DOMINATING SET on weighted AT-free graphs, we obtain the formulas

$$\gamma_{\mathrm{i}}^{w}(G) = \min_{x \in V} \left(w(x) + \sum_{j=1}^{r(x)} \gamma_{\mathrm{i}}^{w}(C_j^x) \right),$$

$$\gamma_{\mathrm{i}}^{w}(C^x) = \min_{y \in C^x} \left(w(y) + \gamma_{\mathrm{i}}^{w}(I(x,y)) + \sum_{j} \gamma_{\mathrm{i}}^{w}(D_j^y) \right),$$

$$\gamma_{\mathrm{i}}^{w}(I) = \min_{s \in I} \left(w(s) + \gamma_{\mathrm{i}}^{w}(I(x,s)) + \gamma_{\mathrm{i}}^{w}(I(s,y)) + \sum_{j} \gamma_{\mathrm{i}}^{w}(C_j^s) \right),$$

analogously to the formulas in Lemmas 7.2, 7.3, and 7.4, respectively. Consequently, there exists an algorithm computing $\gamma_{\mathrm{i}}^{w}(G)$ for a weighted AT-free graph G in time $O(n^4)$.

10. Bounded asteroidal number. In this section we show that the independence number of graphs with bounded asteroidal number can be computed in polynomial time.

DEFINITION 10.1. *The* asteroidal number *of a graph G is the maximum cardinality of an asteroidal set in G.*

Hence a graph is AT-free if and only if its asteroidal number is at most 2. Furthermore, the asteroidal number of a graph G is bounded by $\alpha(G)$, since every asteroidal set is an independent set. Computing the asteroidal number of a graph is NP-complete in general, but solvable in polynomial time for many graph classes [16].

DEFINITION 10.2. *Let Ω be an asteroidal set of G. The* lump $L(\Omega)$ *is the set of vertices v such that for all $x \in \Omega$ there is a component of $G - N[x]$ containing v and $\Omega \setminus \{x\}$.*

Let $\Omega = \{x_1, \ldots, x_\kappa\}$ be an asteroidal set of cardinality $\kappa \geq 2$ and consider the lump $L = L(\Omega)$.

Let s be an arbitrary vertex in L. Now we show how $N[s]$ splits the lump analogously to Theorem 4.4.

Consider the components of $G - N[s]$. These components partition Ω into sets $\Omega_1, \ldots, \Omega_\tau$, where each Ω_i is a maximal subset of Ω contained in a component of $G - N[s]$.

LEMMA 10.3. *For each $i = 1, \ldots, \tau$, the set $\Omega_i^* = \Omega_i \cup \{s\}$ is an asteroidal set in G.*

Proof. Consider $x \in \Omega_i$. Then, by definition, $\Omega \setminus \{x\}$ and s are contained in one component of $G - N[x]$. Hence, $\Omega_i^* \setminus \{x\}$ is contained in one component of $G - N[x]$. This proves the claim. ∎

LEMMA 10.4. *Let $z \in L$ be in some component C^* of $G - N[s]$ that contains no vertices of Ω. Then $C^* \subseteq L$.*

Proof. Let $p \in C^* \setminus \{z\}$. There is a p,z-path avoiding $N[x]$ for any vertex $x \in \Omega$. This proves the claim. ∎

First we consider the case where $\tau = 1$, i.e., where Ω is in one component of $G - N[s]$. Then $\Omega \cup \{s\}$ is an asteroidal set.

LEMMA 10.5. *If Ω is contained in one component C of $G - N[s]$, then $L(\Omega \cup \{s\}) = L \cap C$.*

Proof. Clearly $L(\Omega \cup \{s\}) \subseteq L \cap C$. Let $z \in L \cap C$ and consider a vertex $x \in \Omega$. Clearly, there is an x, z-path avoiding $N[s]$, since z and x are in the component C of $G - N[s]$. Hence z is in the component of $G - N[s]$ containing Ω. Consider any other vertex $y \in \Omega$. (Such a vertex exists since $|\Omega| \geq 2$.) Then there is a z, y-path avoiding $N[x]$ since $z \in L$. Furthermore, there is a y, s-path avoiding $N[x]$ since $\Omega \cup \{s\}$ is an asteroidal set. Hence z is in the component of $(\Omega \cup \{s\}) \setminus \{x\}$ of $G - N[x]$. ∎

Now we consider the case where $\tau > 1$. Let $L_i = L(\Omega_i \cup \{s\})$ for $i = 1, \dots, \tau$. Clearly, $L_i \cap L_j = \emptyset$ for every $i \neq j$.

LEMMA 10.6. *Assume $\tau > 1$ and let C be the component of $G - N[s]$ containing Ω_i. Then $L_i = L \cap C$.*

Proof. First let $z \in L \cap C$. Then for all x and y in Ω_i there is a z, x-path avoiding $N[s]$ since $z \in C$ (showing that z and Ω_i are in one component of $G - N[s]$), and there is a z, x-path avoiding $N[y]$ since $z \in L$. For $y' \in \Omega_j$ for any $j \neq i$ there is a z, y'-path avoiding $N[x]$, since $z \in L$. Such a path contains a vertex of $N[s]$, and consequently there is a z, s-path avoiding $N[x]$. This shows that z, s and $\Omega_i \setminus \{x\}$ are in one component of $G - N[x]$ and hence $L \cap C \subseteq L_i$.

Now let $z \in L_i$. This clearly implies $z \in C$. For a vertex $y \in \Omega_j$, $j \neq i$, s and the set $\Omega \setminus \{y\}$ are in one component of $G - N[y]$ since $s \in L$. There is an s, y-path avoiding $N[y]$ since y and z belong to different components of $G - N[s]$. Consequently, z and $\Omega \setminus \{y\}$ are in one component of $G - N[y]$.

For a vertex $x \in \Omega_i$, there is a component of $G - N[x]$ containing s and $\Omega \setminus \{x\}$, since $s \in L$. Since $z \in L_i$, there is an s, z-path avoiding $N[x]$. Hence also z is in this component of $G - N[x]$ and therefore $L_i \subseteq L \cap C$. ∎

THEOREM 10.7. *There exist components $C_1, \dots, C_t$ of $G - N[s]$ which contain no vertex of Ω such that*

$$L \setminus N[s] = \bigcup_{i=1}^{t} C_i \cup \bigcup_{j=1}^{\tau} L_j.$$

Proof. Let $C_1, \dots, C_t$ be the components of $G - N[s]$ which contain a vertex of L but no vertex of Ω. Then by Lemma 10.4 we have $\bigcup_{i=1}^{t} C_i \subseteq L \setminus N[s]$, and by Lemmas 10.5 and 10.6 we have $\bigcup_{j=1}^{\tau} L_j \subseteq L \setminus N[s]$.

Now let $l \in L \setminus N[s]$. If l is in a component containing Ω_i, $1 \leq i \leq \tau$, then $l \in L_i$ by Lemma 10.5 or 10.6. Otherwise there is an index i, $1 \leq i \leq t$, such that $l \in C_i$. This completes the proof. ∎

Theorem 10.7 enables us to generalize Lemmas 6.3, 7.4, and 8.4 in the following way.

LEMMA 10.8. *Let $L = L(\Omega)$ be a lump of G. If $L = \emptyset$ then $\alpha(L) = \gamma_{\mathrm{i}}(L) = \gamma_{\mathrm{ip}}(\Omega, L) = 0$. Otherwise*

$$\alpha(L) = 1 + \max_{s \in L} \left(\sum_{j=1}^{t} \alpha(C_j) + \sum_{i=1}^{\tau} \alpha(L_i) \right),$$

$$\gamma_{\mathrm{i}}(L) = 1 + \min_{s \in L} \left(\sum_{j=1}^{t} \gamma_{\mathrm{i}}(C_j) + \sum_{k=1}^{\tau} \gamma_{\mathrm{i}}(L_k) \right),$$

$$\gamma_{\mathrm{ip}}(\Omega, L) = 1 + \min_{s \in \Delta(\Omega, L)} \left(\sum_{j=1}^{t} \gamma_{\mathrm{ip}}(s, C_j) + \sum_{k=1}^{\tau} \gamma_{\mathrm{i}}(\Omega_k + s, L_k) \right),$$

where $C_1, \ldots, C_t$ *are the components of* $G - N[s]$ *which contain no vertex of* Ω, $L_1, \ldots, L_\tau$ *are the lumps* $L(\Omega_i + s)$ *as used in Lemma* 10.3, $\Delta(\Omega, L) = \{z \in L \mid d_G(z, \Omega) > 2\}$, *and* $\gamma_{\mathrm{ip}}(\Omega, L)$ *is the minimum cardinality of an independent perfect dominating set of* $G[L]$ *contained in* $\Delta(\Omega, L)$. *Moreover, if* $L \neq \emptyset$ *but* $\Delta(\Omega, L) = \emptyset$, *then* $\gamma_{\mathrm{ip}}(\Omega, L) = \infty$.

Together with Lemmas 6.1 and 6.2, 7.2 and 7.3, and 8.2 and 8.4, the formulas of Lemma 10.8 lead to algorithms computing $\alpha(G)$, $\gamma_{\mathrm{i}}(G)$, and $\gamma_{\mathrm{ip}}(G)$ for a graph G. For any positive integer k, these algorithms can be implemented to run in time $O(n^{k+2})$ for all graphs with asteroidal number at most k. Analogously to the proof of Theorem 6.4, the time complexity is now dominated by the term $\sum_\Omega \sum_{s \in L(\Omega)} O(|L(\Omega)|) = O(n^{k+2})$, where the sum is taken over all asteroidal sets Ω of G and all $s \in L(\Omega)$.

As before, our algorithms for graphs with a bounded asteroidal number can be extended to the weighted cases of the problems and the corresponding algorithms run within the same time bounds.

11. Cliques. Contrary to the independent set problems considered so far, the NP-complete graph problems CLIQUE and PARTITION INTO CLIQUES, that are closely related to INDEPENDENT SET, both remain NP-complete when restricted to the class of AT-free graphs. Concerning CLIQUE, recall that Poljak has shown that INDEPENDENT SET remains NP-complete on triangle-free graphs (see [9]). Consequently CLIQUE remains NP-complete on graphs with independence number at most 2, and thus on AT-free graphs.

Similarly, it follows from a recent result due to Maffray and Preissman (showing that the problem GRAPH k-COLORABILITY remains NP-complete when restricted to triangle-free graphs [22]), that the problem PARTITION INTO CLIQUES remains NP-complete on AT-free graphs.

Therefore CLIQUE and PARTITION INTO CLIQUES are the first NP-complete graph problems known to us which are NP-complete on AT-free graphs, but solvable in polynomial time on the class of cocomparability graphs. The latter graph class is the largest well-studied subclass of AT-free graphs which is also a class of perfect graphs.

12. Conclusions. In this paper we have shown that the maximum weight of an independent set in a weighted AT-free graph can be computed in time $O(n^4)$. The same approach can be used to obtain $O(n^4)$ algorithms to solve the (weighted) INDEPENDENT DOMINATING SET problem and the INDEPENDENT PERFECT DOMINATING SET problem on AT-free graphs. We have also shown how to adapt the algorithm computing the independence number in such a way that the new algorithm computes the independence number of a graph with a bounded asteroidal number in polynomial time.

All our algorithms can be modified such that they not only compute the optimal weight of a set of certain type (e.g., the maximum weight of an independent set) but also a set realizing the optimal weight (e.g., a maximum weight independent set) within the same time bound.

From the current knowledge it would be interesting to find out the algorithmic complexity of the following well-known NP-complete graph problems when restricted to AT-free graphs: GRAPH k-COLORABILITY, HAMILTONIAN CIRCUIT, HAMILTONIAN PATH. These three problems are all known to have polynomial time algorithms for cocomparability graphs [8, 10].

REFERENCES

[1] S. Arnborg, D. G. Corneil, and A. Proskurowski, *Complexity of finding embeddings in a k-tree*, SIAM J. Algebraic Discrete Methods, 8 (1987), pp. 277–284.

[2] H. Balakrishnan, A. Rajaraman, and C. Pandu Rangan, *Connected domination and Steiner set on asteroidal triple-free graphs*, in Proceedings of WADS'93, Lecture Notes in Computer Science 709, Springer-Verlag, New York, 1993, pp. 131–141.

[3] A. Brandstädt, *Special graph classes – A survey*, Schriftenreihe des Fachbereichs Mathematik, SM-DU-199, Universität Duisburg Gesamthochschule, Duisburg, Germany, 1991.

[4] H. Breu and D. G. Kirkpatrick, *Algorithms for Domination and Steiner Tree Problems in Cocomparability Graphs*, University of British Columbia, Vancouver, manuscript 1993.

[5] M. S. Chang, *Weighted domination on cocomparability graphs*, in Proceedings of ISAAC'95, Lecture Notes in Computer Science 1004, Springer-Verlag, New York, 1996, pp. 122–131.

[6] D. G.Corneil, S. Olariu, and L. Stewart, *Asteroidal triple-free graphs*, SIAM J. Discrete Math., 10 (1997), pp. 399–430.

[7] D. G. Corneil, S. Olariu, and L. Stewart, *A linear time algorithm to compute dominating pairs in asteroidal triple-free graphs*, in Proceedings of ICALP'95, Lecture Notes in Computer Science 944, Springer-Verlag, New York, 1995, pp. 292–302.

[8] J. S. Deogun and G. Steiner, *Polynomial algorithms for Hamiltonian cycle in cocomparability graphs*, SIAM J. Comput., 23 (1994), pp. 520–552.

[9] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.

[10] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[11] M. Grötschel, L. Lovász, and A. Schrijver, *Polynomial algorithms for perfect graphs*, Ann. Disc. Math., 21 (1984), pp. 325–356.

[12] J. Hastad, *Clique is hard to approximate within* $n^{1-\epsilon}$, in Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science, Burlington, VT, 1996, pp. 627–636.

[13] D. S. Johnson, *The NP-completeness column: An ongoing guide*, J. Algorithms, 6 (1985), pp. 434–451.

[14] T. Kloks, *Treewidth – Computations and Approximations*, Lecture Notes in Computer Science 842, Springer-Verlag, New York, 1994.

[15] H. J. Broersma, T. Kloks, D. Kratsch, and H. Müller, *A generalization of AT-free graphs and a generic algorithm for solving triangulation problems*, in Proceedings of WG'98, Lecture Notes in Computer Science 1517, Springer-Verlag, New York, 1998, pp. 88–89.

[16] T. Kloks, D. Kratsch, and H. Müller, *Asteroidal sets in graphs*, in Proceedings of WG'97, Lecture Notes in Computer Science 1335, Springer-Verlag, New York, 1997, pp. 229–241.

[17] 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.

[18] T. Kloks, H. Müller, and C. K. Wong, *Vertex ranking of asteroidal triple-free graphs*, in Proceedings of ISAAC'96, Lecture Notes in Computer Science 1178, Springer-Verlag, New York, 1996, pp. 174–182.

[19] D. Kratsch, *Domination and Total Domination on Asteroidal Triple-Free Graphs*, Forschungsergebnisse Math/Inf/96/25, FSU Jena, Germany, 1996.

[20] C. G. Lekkerkerker and J. Ch. Boland, *Representation of a finite graph by a set of intervals on the real line*, Fund. Math., 51 (1962), pp. 45–64.

[21] I. J. Lin, T. A. McKee, and D. B. West, *Leafage of chordal graphs*, Discuss. Math. Graph Theory, 18 (1998), pp. 23–48.

[22] F. Maffray and M. Preissman, *On the NP-completeness of the k-colorability problem for triangle-free graphs*, Discrete Math., 162 (1996), pp. 313–317.

[23] R. H. Möhring, *Triangulating graphs without asteroidal triples*, Discrete Appl. Math., 64 (1996), pp. 281–287.

[24] R. M. McConnell and J. P. Spinrad, *Linear time transitive orientation*, in Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, PA, 1997, pp. 19–25.

[25] E. Prisner, *Representing triangulated graphs in stars*, Abh. Math. Sem. der Univ. Hamburg, 62 (1992), pp. 29–41.

[26] R. Uehara, *NP-Complete Problems on a 3-Connected Cubic Planar Graph and Their Applications*, Technical report TWCU-M-0004, Tokyo Woman's Christian University, 1996.

[27] J. R. Walter, *Representations of chordal graphs as subtrees of a tree*, J. Graph Theory, 2 (1978), pp. 265–267.

[28] M. Yannakakis, *Computing the minimum fill-in is NP-complete*, SIAM J. Algebraic Discrete Methods, 2 (1981), pp. 77–79.

SIAM J. DISCRETE MATH.
Vol. 12, No. 3, pp. 289–297

A 2-APPROXIMATION ALGORITHM FOR THE UNDIRECTED FEEDBACK VERTEX SET PROBLEM*

VINEET BAFNA†, PIOTR BERMAN‡, AND TOSHIHIRO FUJITO§

Abstract. A feedback vertex set of a graph is a subset of vertices that contains at least one vertex from every cycle in the graph. The problem considered is that of finding a minimum feedback vertex set given a weighted and undirected graph. We present a simple and efficient approximation algorithm with performance ratio of at most 2, improving previous best bounds for either weighted or unweighted cases of the problem. Any further improvement on this bound, matching the best constant factor known for the vertex cover problem, is deemed challenging.

The approximation principle, underlying the algorithm, is based on a generalized form of the classical local ratio theorem, originally developed for approximation of the vertex cover problem, and a more flexible style of its application.

Key words. approximation algorithm, performance guarantee, feedback vertex set problem, local ratio theorem

AMS subject classifications. 68Q25, 90C27, 05C85, 05C38

PII. S0895480196305124

1. Introduction.

We are concerned with polynomial time approximation of the feedback vertex set problem on weighted, undirected graphs. A *feedback vertex set* (FVS) of a graph G is a set of vertices such that every cycle in G contains at least one vertex of the set. In a general setting some rational value is associated with each vertex of G as its *weight*. The FVS problem is then defined to be that of finding an FVS of minimum total weight in a given graph. In this paper we always assume the case of *weighted* (i.e., arbitrary weights on vertices) and *undirected* graphs unless otherwise specified.

This problem is of fundamental importance in combinatorial optimization. One typical application appears, as suggested by the name, in the context of combinatorial circuit design. The circuits are represented by graphs in which cycles potentially imply a "racing condition"; that is, some circuit element might receive new inputs before it stabilizes. One way to avoid such a condition is by placing a clocked register at each cycle in the circuit; in that case, we would like to keep the number of clocked registers as low as possible. The minimum FVS for the graph gives a bound on the number of registers needed. For other applications, e.g., in the areas of constraint satisfaction problems and Bayesian inference, see Bar-Yehuda et al. [4].

1.1. Short history and related work.

The FVS problem is NP-hard; for directed graphs Karp showed its NP-completeness even if graphs are unweighted [12], and essentially the same transformation shows that it is equally hard for undirected

*Received by the editors June 12, 1996; accepted for publication (in revised form) November 23, 1998; published electronically September 7, 1999. A preliminary version of this paper was presented at the Sixth Annual International Symposium on Algorithms and Computation, Cairns, Australia, 1995.

http://www.siam.org/journals/sidma/12-3/30512.html

†DIMACS Center, Piscataway, NJ 08854 (bafna@dimacs.rutgers.edu). The work of this author was supported by Special Year National Science Foundation grant BIR-9412594.

‡Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802 (berman@cse.psu.edu).

§ Department of Electrical Engineering, Hiroshima University, 1-4-1 Kagamiyama, Higashi-Hiroshima 739-8527 Japan (fujito@huis.hiroshima-u.ac.jp).

graphs. Given this intractability it is natural to consider a next best approach: a polynomial time algorithm for computing a near optimal FVS. The quality of an approximation algorithm is measured by its *performance ratio*: the worst case ratio of weight of an approximate solution computed by the algorithm to the optimal solution weight. An algorithm with performance ratio r is also called an r*-approximation* algorithm.

The first nontrivial (i.e., better than $|V| = n$) approximation ratio of $2 \log n$ for unweighted graphs appeared in the early work of Erdős and Pósa [6], where they studied the number of (vertex) disjoint cycles in a graph. It was later improved to $\sqrt{\log n}$ by Monien and Schulz [14], who considered and compared various approaches to the problem. Only recently, Bar-Yehuda et al. were able to show that the smallest cardinality FVS (i.e., *unweighted* version) can be approximated within a constant factor of 4 [4]. Moreover, they considered the weighted version as well and obtained a performance ratio of $\min\{4 \log n, 2\Delta^2\}$, where Δ is the maximum vertex degree of a graph.

As for a lower bound on the performance ratio, the problem is known to be *MAX SNP*-hard [13, 16], implying that the ratio cannot go down arbitrarily close to 1 unless $P = NP$ [1]. In fact, a more direct implication is available due to the fact that the vertex cover (VC) problem is reducible to the FVS problem in an approximation preserving manner [13], so that any performance ratio r for the FVS problem would imply the same ratio r for the VC problem. A better approximation of the VC problem has been a subject of extensive research over the years, yet the best constant approximation ratio has remained at 2. (The overall best one is $2 - \log \log n / 2 \log n$ [3, 15].) On the other hand, a lower bound on the performance ratio for the VC problem has been continuously improved in the last few years, and currently it is known to be NP-hard to guarantee a factor of $7/6 - \epsilon$ for any $\epsilon > 0$ [11], implying the same bound for the FVS problem.

The FVS problem (or feedback *edge* set problem) for *directed* graphs, largely due to more versatile nature of its applicability, has drawn even more attention in various areas. It appears, however, that the problem is harder to approximate, with $O(\log n \log \log n)$ being the best ratio known today [7].

1.2. Our contributions.

Factor-2 approximation. We will present an approximation algorithm for the weighted FVS problem (in section 3) and show that its performance ratio is bounded above by 2 (in section 4), improving upon the previous best of $\min\{4 \log n, 2\Delta^2\}$. Independently of our work, Becker and Geiger have recently discovered a different 2-approximation algorithm [5], their analysis of which is more complicated than ours, without any elucidation of underlying approximation principles. In light of the facts mentioned earlier concerning the approximability of the VC problem and its reducibility to the FVS problem, achieving a better performance ratio, if at all possible, is deemed quite challenging. Our algorithm is also quite simple and efficient; it can be implemented to run in time $O(\min\{|E| \log |V|, |V|^2\})$.

Generalized local ratio approximation. Our approximation method is based on the *local approximation* principle. In a most simple form it was used already in the Gavril's maximal matching–based approximation for the unweighted VC problem [9, p. 134], and later it was explicitly formalized as the *local ratio theorem* by Bar-Yehuda and Even [3].

This principle, however, has been known mostly only in a doubly limited form;

the formulation allowed extraction of only *uniformly* weighted subgraphs from an arbitrarily weighted graph and, as its name suggests, allowed its application only to subgraphs of *small* size, e.g., short odd cycles for VC approximation [3] and short cycles for weighted FVS approximation [4]. An easy but crucial observation presented in this paper is that neither restriction is necessary, leading to a generalization of the theorem (section 2) and demonstration of the effectiveness resulting from more flexible applications of it. Moreover, the principle is applicable not only to the VC or FVS problems but also to other weighted optimization problems of *covering* type, and hence this approximation technique could be of independent interest. For a simpler presentation, however, we restrict ourselves to the FVS problem in this paper.

1.3. Definitions and notation. We use the following definitions and notation throughout the paper. For any graph G let $V(G)$ denote the vertex set of G and $G[U]$ the subgraph of G induced by U for $U \subseteq V(G)$. A (vertex) weighted graph G with a weight function $w : V(G) \to \boldsymbol{Q}_+$ is denoted as (G, w), where a nonnegative rational $w(u)$ represents a weight associated with each vertex u of G. The sum of weights of vertices in U is denoted by $w(U)(= \sum_{u \in U} w(u))$. A collection $\{(G_i, w_i)\}$ of weighted graphs is called a *decomposition* of (G, w) if G_i is a subgraph of G $\forall i$ and $\sum_{i:u \in V(G_i)} w_i(u) \leq w(u)$ $\forall u \in V$. A weight function $w : V \to \boldsymbol{Q}_+$ is called *degree-proportional* if, for some constant $c > 0$, $w(u) = c(d(u) - 1)$ for every $u \in V$, where $d(u)$ denotes degree of u. A graph is called *clean* if it contains no vertex of degree less than 2, and a cycle C is *semidisjoint* if, for every vertex u of C, $d(u) = 2$ with at most one exception. Let $\mathrm{opt}(G, w)$ denote any optimal FVS in (G, w). An FVS F is *minimal* in G if no smaller FVS is contained in F or, equivalently, if $F - \{u\}$ is not an FVS in G for all $u \in F$.

2. Generalization of the local ratio theorem. The local ratio approximation, in its most elementary use, is based on the following principle: If an (*unweighted*) structure G contains a substructure H such that every optimal solution occupies a large portion of H, one can afford to take the whole of H into a solution and reduce the problem by removing H from G. More specifically and when G is a weighted graph, this idea can be implemented by the following operations:

1. Choose a suitable subgraph H (i.e., local structure) of usually small size.
2. "Subtract" H with a *uniform* weight distribution on it from G (i.e., entire structure).
3. Accept into a solution all the vertices of weight *reduced to zero*.

We extend this technique in two directions, one in its formulation and the other in its application. First it will be allowed to include *nonuniformly* weighted subgraphs in our target local structures. This change enables us to choose not only a subgraph but also a weight distribution on it to be sliced off from the whole distribution. Formally, we have the following theorem.

THEOREM 2.1. *Let $\{(G_i, w_i)\}$ be a decomposition of (G, w) and F be any FVS in G such that $w(F) = \sum_i w_i(F \cap V(G_i))$. Then,*

$$\frac{w(F)}{w(opt(G, w))} \leq \max_i \left\{ \frac{w_i(F \cap V(G_i))}{w_i(opt(G_i, w_i))} \right\}$$

Proof. Since $\{(G_i, w_i)\}$ is a decomposition of (G, w), $w(X) \geq \sum_i w_i(X \cap V(G_i))$ for any set $X \subseteq V$. Thus, using the assumption on $w(F)$,

$$\frac{w(F)}{w(\mathrm{opt}(G, w))} \leq \frac{\sum_i w_i(F \cap V(G_i))}{\sum_i w_i(\mathrm{opt}(G, w) \cap V(G_i))}.$$

Besides, if F is an FVS in G, so is its restriction $F \cap V(G_i)$ in every subgraph G_i. Therefore, we have $w_i(\mathrm{opt}(G, w) \cap V(G_i)) \geq w_i(\mathrm{opt}(G_i, w_i))$ for all i, which gives

$$\frac{\sum_i w_i(F \cap V(G_i))}{\sum_i w_i(\mathrm{opt}(G, w) \cap V(G_i))} \leq \frac{\sum_i w_i(F \cap V(G_i))}{\sum_i w_i(\mathrm{opt}(G_i, w_i))}.$$

Since every summand appearing in the fraction on the right-hand side is nonnegative, it can be bounded above by $\max_i\{w_i(F \cap V(G_i))/w_i(\mathrm{opt}(G_i, w_i))\}$. □

In this way the ratio of an FVS weight to the optimal one in (G, w) can be reduced to the ones in its subgraphs (G_i, w_i).

For a subgraph G_1 of G let $\gamma \stackrel{\mathrm{def}}{=} \min\{w(v) : v \in V(G_1)\}$, w_1, be a function on $V(G_1)$ whose value is constantly γ, $w_2 \stackrel{\mathrm{def}}{=} w - w_1$, and let V_0 be a set of vertices u with $w_2(u) = 0$. Then $\{(G_1, w_1), (G[V - V_0], w_2)\}$ is easily a decomposition of (G, w) and, for any FVS F_2 for $G[V - V_0]$, $F \stackrel{\mathrm{def}}{=} V_0 \cup F_2$ is an FVS for G. Let c^* denote the *cardinality* of an optimal FVS for unweighted G_1. The original local ratio theorem of Bar-Yehuda and Even [3] states that the approximation ratio of F is bounded by

$$\max\left\{\frac{|V(G_1)|}{c^*}, \frac{w_2(F_2)}{w_2(\mathrm{opt}(G[V - V_0], w_2))}\right\},$$

and this follows easily from Theorem 2.1 since $w_1(F \cap V(G_1)) \leq w_1(V(G_1)) = \gamma|V(G_1)|$; thus $w_1(F \cap V(G_1))/w_1(\mathrm{opt}(G_1, w_1)) \leq \gamma|V(G_1)|/\gamma c^* = |V(G_1)|/c^*$ and $F \cap V(G_2) = F \cap (V - V_0) = F_2$.

The second extension of the local ratio technique will be demonstrated in the next section, where our algorithm slices up a weight distribution from the entire structure.

3. Approximation algorithm. Our algorithm, called FEEDBACK, is presented in Figure 3.1, where text in square brackets are comments used for analysis only.

Given a graph (G, w) with $G = (V, E)$, any vertex of weight zero is removed from G and placed in the solution set F at the outset. FEEDBACK then decomposes (G, w) into subgraphs (G_i, w_i)'s (in the first While loop) by iteratively subtracting w_i from w, removing vertices of weight reduced to zero, adding them into F, and cleaning up G (by procedure `Cleanup`, which recursively deletes vertices of degree ≤ 1), until G becomes empty.

The subgraph G_i derived in the ith iteration is either a semidisjoint cycle C contained in G or, otherwise, G itself. Note that the first case has precedence over the second; that is, G_i is a semidisjoint cycle whenever G contains one. When G_i is a cycle C it is uniformly weighted with $w_i(u) = \gamma = \min\{w(u) : u \in V(C)\}$, the minimum weight on C, for all $u \in G_i$. Otherwise, G_i is clean and degree-proportionally weighted with $w_i(u) = \gamma(d(u)-1)\ \forall u \in V$. In either case the value of γ is determined such that w_i is maximal without exceeding w, and hence some vertex u of G necessarily has its weight $w(u)$ reduced to zero when w_i is subtracted from w. Such vertices are removed from G, making progress toward emptying G, and at the same time we collect them all in F. The sole purpose of using an auxiliary stack data structure, `STACK`, is to keep track of the (reverse) order in which these vertices are added into F.

The graph G eventually becomes empty (in at most $|V|$ iterations). At this point (i.e., right after the first While) every vertex was swept out in the process, or otherwise it is kept in F. Observe that F is indeed an FVS for the original G because any vertex was cleaned up only after it was found to be useless.

The second While loop examines vertices of this F, one by one, in the reverse order of their inclusion into F. Whenever a vertex is found to be extraneous, it is

Input: an undirected graph $G = (V, E)$ with vertex weights $w : V \to \boldsymbol{Q}_+$
Output: a feedback vertex set F

Initialize $F = \{u \in V : w(u) = 0\}, V = V - F$. $[i = 0]$
`Cleanup`(G)
While $V \neq \emptyset$ do
    $[i \leftarrow i + 1]$
    If G contains a semidisjoint cycle C, then
        Let $\gamma \leftarrow \min\{w(u) : u \in V(C)\}$.
        Set $w(u) \leftarrow w(u) - \gamma, \ \forall u \in V(C)$.
            $[G_i = C$ and $w_i(u) = \gamma, \ \forall u \in V(C)]$
    Else [G is clean and contains no semidisjoint cycle]
        Let $\gamma \leftarrow \min\{w(u)/(d(u) - 1) : u \in V\}$.
        Set $w(u) \leftarrow w(u) - \gamma(d(u) - 1), \ \forall u \in V$.
            $[G_i = G$ and $w_i(u) = \gamma(d(u) - 1), \ \forall u \in V]$
    For each $u \in V$ with $w(u) = 0$ do
        Remove u from V, add it to F, and `push` it onto `STACK`.
    `Cleanup`(G)
While `STACK` $\neq \emptyset$ do
    Let $u \leftarrow$ `pop`(`STACK`).
    If $F - \{u\}$ is an FVS in *original* G, then [u is redundant]
        Remove u from F.

`Cleanup`(G):
    While G contains a vertex of degree at most 1, remove it along with
        any incident edges.

FIG. 3.1. 2-*approximation algorithm* FEEDBACK *for the FVS problem.*

discarded from F. As will be seen later, this process ensures not only that F is a minimal FVS in original G but also that $F \cap V(G_i)$ in G_i $\forall i$.

Running time. The running time of FEEDBACK is dominated by the first While loop. All the operations of cleaning up vertices (along with edges), detecting semi-disjoint cycles, computing the minimum weights on them, and deleting them from a graph can be done in time $O(|V|+|E|)$ by maintaining a collection of existent (disjoint) paths consisting solely of degree-2 vertices. Since each iteration takes $O(|V|)$ time for other operations and there are at most $|V|$ of them, the running time is $O(|V|^2)$.

Alternatively, we may maintain the value of $w(u)/(d(u) - 1)$ for each $u \in V$ in a priority queue P, instead of *individual* vertex weights. The computation of $\gamma = \min\{w(u)/(d(u) - 1) : u \in V\}$ is then supported by the `Extract-Min` operation ($O(\log |V|)$). There are two types of updates for these values: one by subtraction of w_i from w and the other caused by decrement of degrees. In the former case, the new value, after subtraction of w_i, becomes $(w(u) - \gamma(d(u) - 1))/(d(u) - 1) = w(u)/(d(u) - 1) - \gamma$, the old value less γ for each vertex. Thus, the actual value for any vertex in V can be recovered from a sequence of γ values, without changing key values stored in P. For the second case, however, some key values must be changed, but only for those vertices adjacent to u, for each removal of vertex u. We do so using both `Insert` and `Delete` operations, each of which takes $O(\log |V|)$ by the standard implementations, and $O(1)$ time calculation of a new value. Since key values of elements in P need to be modified at most $|E|$ times, it takes $O(|E| \log |V|)$ overall, better than $O(|V|^2)$ when a graph is sparse.

4. Performance ratio. To avoid any possible ambiguity in the following argument, let us fix an input graph $(G = (V, E), \bar{w})$ and the output FVS $\bar{F}$. Recall that our goal is to achieve a globally good approximation ratio by ensuring a good ratio locally at every derived subgraph in a decomposition of the given graph, and our algorithm FEEDBACK is designed exactly to do so. Toward this end it will be shown below, in this order, that the following hold:

1. FEEDBACK computes $\bar{F}$ and a decomposition $\{(G_i, w_i)\}$ of $(G, \bar{w})$ such that

(i) $\bar{w}(\bar{F}) = \sum_i w_i(\bar{F} \cap V(G_i))$ (Lemma 4.1, precondition for application of Theorem 2.1), and

(ii) $\bar{F} \cap V(G_i)$ is a minimal FVS in G_i for all i (Lemma 4.2).

2. The weight of any minimal FVS in any clean, degree-proportionally weighted graph without semidisjoint cycles is small relative to the optimal weight (Lemmas 4.3 and 4.4).

LEMMA 4.1. $\bar{w}(\bar{F}) = \sum_i w_i(\bar{F} \cap V(G_i))$.

Proof. Recall that any vertex u of G can enter $\bar{F}$ only after its weight $w(u)$ is reduced completely to zero by a sequence of subtractions, and hence partial weights $w_i(u)$'s must sum up to the total $\bar{w}(u)$ for any u in $\bar{F}$. (On the other hand, $V - \bar{F}$ consists of those vertices cleaned up from G *prematurely* with nonzero weights). ☐

LEMMA 4.2. *$\bar{F} \cap V(G_i)$ is a minimal FVS in G_i $\forall i$.*

Proof. Let G^i denote the graph remaining right after the $(i-1)$st iteration of the first While is completed. We first claim that $\bar{F} \cap V(G^i)$ is a minimal FVS in G^i. Let F^∞ denote the FVS constructed by the entire run of the first While. Since vertices in F^∞ are examined, in the second While, in the reverse order of their addition to F^∞, all the vertices in $F^\infty \cap V(G^i)$ are tested for their redundancy before those in $F^\infty - V(G^i)$. Let F^i denote the FVS obtained from F^∞, during the second While, by removing any vertices in $F^\infty \cap V(G^i)$ which are redundant in G. Then, $F^i \cap V(G^i)$ must be a minimal FVS in G^i since otherwise, i.e., if $(F^i \cap V(G^i)) - \{u\}$ is an FVS in G^i for some $u \in F^i \cap V(G^i)$, $F^i - \{u\}$ would be an FVS in G, a contradiction. Since $F^i \cap V(G^i) = \bar{F} \cap V(G^i)$, the claim follows.

It remains to observe that if $\bar{F} \cap V(G^i)$ is a minimal FVS in G^i, so is $\bar{F} \cap V(G_i)$ in G_i. Recall that G_i is chosen such that either $G_i = G^i$ or $G_i = C$, some semidisjoint cycle contained in G^i. It is obvious when $G_i = G^i$, and now suppose $G_i = C$. Then, $\bar{F} \cap G_i$ contains *exactly* one vertex of G_i, and hence minimal in G_i, because any minimal FVS in G^i, such as $\bar{F} \cap V(G^i)$, can contain only one vertex from semidisjoint C. ☐

We next consider how large the weight of $\bar{F}$ is when estimated in subgraphs (G_i, w_i)'s, relative to the optimal weights for them. Recall that (G_i, w_i) is in the form of either

1. a simple cycle of identically weighted vertices, or
2. a clean and degree-proportionally weighted graph containing no semidisjoint cycles.

In the first case, the minimality of FVS $\bar{F} \cap V(G_i)$ in G_i actually implies its optimality in G_i. The second case is more interesting. We show that, in this case, the weight of any minimal FVS is bounded above by twice the optimum weight.

Clearly, it suffices to prove this only for the case when $w(u) = d(u) - 1$ for every $u \in V$; this is assumed below in Lemmas 4.3 and 4.4. We will also use a *potential* function $p : V \to \boldsymbol{Q}$ defined as $p(u) = d(u)/2 - 1$ for every $u \in V$. Let $p(U)$ denote

$\sum_{u \in U} p(u)$ for any $U \subseteq V$.

LEMMA 4.3. *For an arbitrary FVS F in $G = (V, E)$, $w(F) \geq p(V) + 1$.*

Proof. Let $\delta(F)$ denote the set of edges incident to some vertex in F. Then, since $\sum_{u \in F} d(u) \geq |\delta(F)|$, $w(F) = \sum_{u \in F}(d(u) - 1) = \sum_{u \in F} d(u) - |F| \geq |\delta(F)| - |F|$. Besides, F is an FVS. Thus $G[V - F]$, with its edge set being $E - \delta(F)$, is acyclic, containing at most $|V - F| - 1$ edges. That is, $|E - \delta(F)| \leq |V - F| - 1$ and hence $w(F) \geq |E| - |V| + 1 = p(V) + 1$ since $p(V) = \sum_{u \in V}(d(u)/2 - 1) = |E| - |V|$. ∎

Observe now that if F is a minimal FVS, each vertex of F is *blocked* by a tree in the forest $G[V - F]$; i.e., each vertex $u \in F$ is joined via two edges to some tree T in $G[V - F]$. Let e_T be the number of edges with one end in T and the other in F. Suppose that T has t vertices. Then $\sum_{u \in V(T)} d(u) = e_T + 2(t - 1)$. Consequently,

$$\begin{aligned} p(V(T)) &= \sum_{u \in V(T)} \left(\frac{d(u)}{2} - 1 \right) \\ &= \frac{\sum_{u \in V(T)} d(u)}{2} - t = \frac{e_T + 2(t-1)}{2} - t = \frac{e_T}{2} - 1. \end{aligned} \tag{4.1}$$

LEMMA 4.4. *If F is a minimal FVS in a clean graph $G = (V, E)$ without a semidisjoint cycle, then $w(F) \leq 2p(V)$.*

Proof. Each vertex $u \in F$ is a priori allocated a potential of $(d(u) - 2)/2$. We show that each $u \in F$ also receives an additional potential of $1/2$ from vertices in $V - F$. Let T be a tree blocking u (so $e_T \geq 2$). Notice that $e_T = 2$ would imply either G is not clean or G contains a semidisjoint cycle; hence $e_T \geq 3$. Also, the total number of vertices each tree T' in $G[V - F]$ can block is at most $\lfloor e_{T'}/2 \rfloor$. Thus, using (4.1), an extra potential of $(e_T/2 - 1)/\lfloor e_T/2 \rfloor$ can be shipped to every vertex u of F from $V - F$, which is at least $1/2$ when $e_T \geq 3$. Therefore, u contributes $d(u) - 1$ to $w(F)$ and at least $(d(u) - 1)/2$ to $p(V)$. ∎

Lemmas 4.3 and 4.4 jointly assert that when G_i is a clean and degree-proportionally weighted graph without semidisjoint cycles, since $\bar{F} \cap V(G_i)$ is a minimal FVS in G_i (Lemma 4.2), the ratio $w_i(\bar{F} \cap V(G_i))/w_i(\text{opt}(G_i, w_i))$ is bounded by $2p(V(G_i))/(p(V(G_i)) + 1)$.

THEOREM 4.5. *The algorithm* FEEDBACK *finds an FVS $\bar{F}$ in $(G, \bar{w})$, where $G = (V, E)$, with approximation factor of $2 - 2/(|E| - 3)$ in time $O(\min(|E| \log |V|, |V|^2))$.*

Proof. Apply Theorem 2.1 using $\{(G_i, w_i)\}$ computed (implicitly) by FEEDBACK as a decomposition of $(G, \bar{w})$. As observed above, when G_i is a uniformly weighted simple cycle (case 1), the local ratio $w_i(\bar{F} \cap V(G_i))/w_i(\text{opt}(G_i, w_i)) = 1$. On the other hand, when G_i is a clean graph without semidisjoint cycles, it is bounded by

$$\frac{2p(V(G_i))}{p(V(G_i)) + 1} = 2 - \frac{2}{p(V(G_i)) + 1} = 2 - \frac{2}{|E(G_i)| - |V(G_i)| + 1} \leq 2 - \frac{2}{|E| - 3}$$

since G_i must contain at least four vertices. ∎

Additionally, it can be seen that the analysis above is essentially tight: there is an infinite sequence of graphs for which the approximation factor of FEEDBACK approaches arbitrarily close to 2. Consider, e.g., a graph G consisting of k triangles $\{a_i, b_i, c_i\}$, $i = 1, \ldots, k$, which are chained together by edges $\{b_i, a_{i+1}\}$, $i = 1, \ldots, k-1$, and $\{b_k, a_1\}$ (see Figure 4.1). Suppose now that G is degree-proportionally weighted, e.g., $w(a_i) = w(b_i) = 2$, and $w(c_i) = 1$, $\forall i$. The set $A = \{a_i : 1 \leq i \leq k\}$ is a minimal FVS with $w(A) = 2k$. On the other hand, an optimal FVS could be formed by c_i's, $1 \leq i \leq k - 1$, plus a_1, with its weight totaling $(k - 1) + 2 = k + 1$.

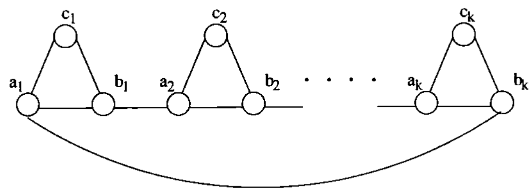


FIG. 4.1. *Example with ratio* = 2.

5. Final remarks. In this paper we have presented a simple and efficient approximation algorithm for the FVS problem on undirected graphs, with a performance ratio of at most 2. While this ratio matches the best constant approximation factor known for the VC problem, there still exists a small gap from the overall best of $2 - \log\log n/2\log n$ for VC. Also, some related directions for further research are suggested when it is taken into account that VC, when restricted to some special classes of graphs, becomes polynomially solvable or easier to approximate. For instance, one of the well-studied cases occurs when graphs are restricted to be of small vertex degree, for which currently the best-known bound is $2 - 3/(\Delta + 2)$ [10], where Δ is the maximal vertex degree. Although the technique introduced in the paper alone is not powerful enough to do any better for this special case, it can be shown that, when combined with other approximation preserving reductions, it yields the performance ratio of $2 - 2/(3\Delta - 2)$. (Interested readers are referred to [2].) It remains an open question whether one can approximate the FVS problem *exactly* as well as the VC problem can be, whether the degree is bounded or not.

Other interesting questions for further investigation include applicability of the techniques developed herein to other problems. For instance, see [8], where, inspired by our work, a similar approach was employed and shown to be effective in approximation of other *node-deletion* problems as well.

Acknowledgment. We gratefully acknowledge a number of valuable comments and suggestions given by the anonymous referees.

REFERENCES

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, *Proof verification and hardness of approximation problems*, in 33rd Annual Symposium on Foundations of Computer Science, Pittsburgh, PA, 1992, pp. 14–23.

[2] V. Bafna, P. Berman, and T. Fujito, *Constant Ratio Approximations of Feedback Vertex Sets in Weighted Undirected Graphs*, Tech. Report TR-96-29, DIMACS, Piscataway, NJ, 1996.

[3] R. Bar-Yehuda and S. Even, *A local-ratio theorem for approximating the weighted vertex cover problem*, in Ann. Discrete Math. 25, North–Holland, Amsterdam, 1985, pp. 27–46.

[4] R. Bar-Yehuda, D. Geiger, J. Naor, and R. M. Roth, *Approximation algorithms for the feedback vertex set problem with applications to constraint satisfaction and Bayesian inference*, SIAM J. Comput., 27 (1998), pp. 942–959.

[5] A. Becker and D. Geiger, *Approximation algorithms for the loop cutset problem*, in Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence, San Francisco, CA, Morgan Kaufman, 1994, pp. 60–68.

[6] P. Erdős and L. Pósa, *On the maximal number of disjoint circuits of a graph*, Publ. Math. Debrecen, 9 (1962), pp. 3–12.

[7] G. Even, J. Naor, B. Schieber, and M. Sudan, *Approximating minimum feedback sets and multi-cuts in directed graphs*, in Fourth Integer Programming and Combinatorial Opti-

mization, Lecture Notes in Comput. Sci. 920, Springer-Verlag, New York, 1995, pp. 14–28.
[8] T. Fujito, *A unified approximation algorithm for node-deletion problems*, Discrete Appl. Math., 86 (1998), pp. 213–231.
[9] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, 1979.
[10] M. Halldórsson and H. Lau, *Low-degree graph partitioning via local search with applications to constraint satisfaction, max cut, and coloring*, J. Graph Algorithms Appl., 1 (1997), pp. 1–13.
[11] J. Håstad, *Some optimal in-approximability results*, in Proceedings of the 29th Annual ACM Symposium on Theory of Computing, El Paso, TX, 1997, pp. 1–10.
[12] 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.
[13] J. Lewis and M. Yannakakis, *The node-deletion problem for hereditary properties is NP-complete*, J. Comput. System Sci., 20 (1980), pp. 219–230.
[14] B. Monien and R. Schulz, *Four approximation algorithms for the feedback vertex set problem*, in Proceedings of the Seventh Conference on Graph Theoretic Concepts of Computer Science, Munich, Hanser Verlag, 1981, pp. 315–326.
[15] B. Monien and E. Speckenmeyer, *Ramsey numbers and an approximation algorithm for the vertex cover problem*, Acta Inform., 22 (1985), pp. 115–123.
[16] C. Papadimitriou and M. Yannakakis, *Optimization, approximation and complexity classes*, J. Comput. System Sci., 43 (1991), pp. 425–440.

SIAM J. DISCRETE MATH.
Vol. 12, No. 3, pp. 298–306

SWITCHING DISTANCE BETWEEN GRAPHS WITH THE SAME DEGREES*

TODD G. WILL†

Abstract. A *switching* replaces a pair of edges ab, cd of a graph G with the pair ac, bd of $\overline{G}$. If G and H have the same degree sequence, then G can be transformed into H by a sequence of switchings. We show that finding the minimum number of switchings required to transform G into H is equivalent to finding the maximum size of a *symmetric circuit partition* of $G\Delta H$. Computing this latter quantity is shown to be NP-complete.

Key words. graph, switching, degree sequence, NP-complete

AMS subject classifications. 05C12, 68Q25

PII. S0895480197331156

1. Introduction.

In a simple graph, replacing a pair of edges ab, cd of G with the pair ac, bd of $\overline{G}$ is called a *switching*, since it switches a pair of edges for a pair of nonedges. Notice that a switching does not change the vertex degrees of G. We say that simple graphs G and H *have the same degrees* if they share the same vertex set V and, $\forall v \in V$, $d_G(v) = d_H(v)$.

Given any graph G, $\Gamma(G)$ is the graph with a vertex corresponding to each graph with the same degrees as G, with two vertices adjacent if and only if the corresponding graphs differ by a single switching [1]. It is not difficult to show that for any simple graph G, $\Gamma(G)$ is connected [6]. Majcher extended this result to multigraphs and provided a procedure for finding a path (i.e., a sequence of switchings) between any two (multi-)graphs with the same degrees [3]. For simple graphs, Taylor has shown that the subgraph of $\Gamma(G)$ induced by vertices corresponding to connected or 2-connected graphs is itself connected [4, 5].

For graphs G and H with the same degrees, we define the *switching distance* between G and H, $d(G, H)$ to be the distance between G and H in $\Gamma(G)$. In other words, $d(G, H)$ is the minimum number of switchings required to transform G into H. In this paper we relate $d(G, H)$ to an edge-partition problem and show that computing $d(G, H)$ is NP-complete.

Given two graphs G and H on the same vertex set V, let $G \cup H$ and $G - H$ denote the graphs on V with edge sets $E(G) \cup E(H)$ and $E(G) \setminus E(H)$, respectively. The *symmetric difference graph* $G\Delta H$ can then be defined as $(G - H) \cup (H - G)$. A *circuit* is a positive length closed walk which traverses any edge at most once; however, it differs from a cycle in that it may use a vertex more than once. We define a *symmetric circuit* in $G\Delta H$ as a circuit whose edges alternate between $G - H$ and $H - G$. A *symmetric circuit partition* of $G\Delta H$ is a set of pairwise edge-disjoint symmetric circuits using all the edges of $G\Delta H$. For graphs G and H with the same degrees, let $m(G\Delta H)$ be the maximum number of circuits in any symmetric circuit partition of $G\Delta H$. The following lemma shows that $m(G\Delta H)$ is well defined.

LEMMA 1.1. *If G and H have the same degrees, then $G\Delta H$ can be partitioned into edge-disjoint symmetric circuits.*

---

*Received by the editors December 5, 1997; accepted for publication (in revised form) November 23, 1998; published electronically September 7, 1999.
http://www.siam.org/journals/sidma/12-3/33115.html
†Mathematics Department, Davidson College, Davidson, NC 28036-1719 (towill@davidson.edu).

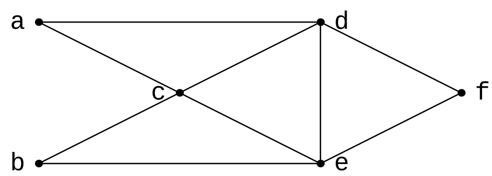


FIG. 2.1.

Proof. Color the edges in $E(G) - E(H)$ blue and the edges in $E(H) - E(G)$ red. The key fact is that for any vertex v, the number of blue edges incident to v is the same as the number of red edges incident to v. This is because $d_G(v) = d_H(v)$ and the only edges of G or H not appearing in $G\Delta H$ are those appearing in both G and H. Thus, we may start at an arbitrary vertex x and begin a trail which alternates color. When we enter any vertex other than x on a edge of a given color, there must be an edge of the opposite color to exit on. Hence, we can continue along any alternating trail until we return to our starting point on a color opposite the one we first left on. After removing this circuit, each vertex is still incident to the same number of red and blue edges. Hence we can use this symmetric circuit in the partition, and partition the remaining edges by induction. □

The main results of this paper show that if G and H have the same degrees, then $d(G, H) = \frac{1}{2}|E(G\Delta H)| - m(G\Delta H)$, and that computing $m(G\Delta H)$ is NP-complete.

2. Relating $d(G, H)$ and $m(G\Delta H)$. We view a graph G as a two-coloring of the complete graph with edges of G colored blue and edges of $\overline{G}$ colored red. An *alternating circuit* is a circuit whose edges alternate between red and blue. Reversing an alternating circuit in G means interchanging the coloring on the circuit and leaving fixed the colors of the edges not in the circuit. In this language, a switching is a transformation that reverses an alternating circuit of length 4.

Note that if C is a symmetric circuit in $G\Delta H$, then C is an alternating circuit in G (and H as well). Reversing C in G (or H) removes C from $G\Delta H$, thus bringing G and H closer together. Because of this connection, we are interested in the number of switchings required to reverse an alternating circuit.

First we fix some notation. We specify a circuit C by listing its vertices in cyclic order and write $xy \in C$ to indicate that x and y occur consecutively in the sequence in either order. For any cyclic sequence $(v_1, \ldots, v_n, v_1)$, let $[i, j]$ be the sequence $(v_i, v_{i+1}, \ldots, v_j)$ and $[i, j]^+$ be the cyclic sequence $(v_i, v_{i+1}, \ldots, v_j, v_i)$.

The results that we prove for circuits are much easier to prove for cycles. For this reason we would like to measure how far a circuit is from being a cycle. To this end, given a circuit $C = (v_1, \ldots, v_s, v_1)$, for each index i let $D(i)$ be the set of indices $j \notin \{i-1, i, i+1\}$ such that $v_j = v_i$ or $v_j v_i \in C$. For example, consider the circuit $C = (v_1, \ldots, v_9, v_1) = (b, c, a, d, c, e, d, f, e, b)$ in the graph shown in Figure 2.1. For this circuit, $v_1 = b, v_2 = v_5 = c, v_3 = a, v_4 = v_7 = d, v_6 = v_9 = e, v_8 = f$, and $D(1) = \{5, 6\}$, $D(2) = \{4, 5, 6, 7, 9\}$.

LEMMA 2.1. *Let $C = (v_1, \ldots, v_{2s}, v_1)$ be an alternating circuit C. If there exist indices $j > i + 1$ of opposite parity such that $j \notin D(i)$, then reversing C can be accomplished by reversing two shorter alternating circuits of length $j - i + 1$ and $2s - j + i + 1$.*

Proof. The hypotheses $j > i + 1$ and $j \notin D(i)$ imply $v_i v_j \notin C$. Regardless

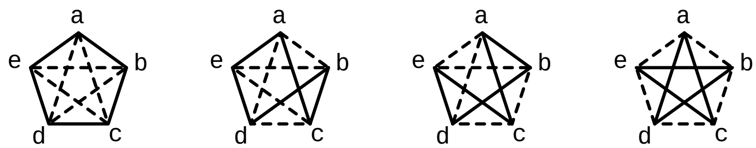


FIG. 2.2.

of the color of v_iv_j, exactly one of the sequences $[i,j]^+$ and $[j,i]^+$ is an alternating circuit. Reversing this circuit changes the color of v_iv_j, making the other sequence an alternating circuit. Reversing the second circuit restores the original color to v_iv_j and completes the reversal of C. One of the circuits has length $j-i+1$ and the other $2s-(j-i)+1$. ∎

LEMMA 2.2. *Reversing an alternating circuit of length $2s$ requires at most $s-1$ switchings.*

Proof. The proof is by induction on s. Let $C=(v_1,\ldots,v_{2s},v_1)$ be an alternating circuit of length $2s$. For $s=2$, the reversal of C is a single switching, so suppose $s>2$. Suppose there exist indices $i<j$ of the same parity with $v_i=v_j$. We can partition the edges of C into two shorter alternating circuits, $[i,j]$ and $[j,i]$ of lengths $j-i$ and $2s-j+i$, respectively. By the induction hypothesis each of these can be reversed, using a total of at most $(\frac{1}{2}(j-i)-1)+(s+\frac{1}{2}(i-j)-1)$ switchings. Thus C can be reversed by a sequence of at most $s-2$ switchings. Henceforth we assume that each vertex appears at most twice in C, once with an index of each parity.

In particular this implies that if a vertex v_i appears exactly once in C, then $\forall j\in D(i)$, v_j must be the opposite parity occurrence of either v_{i-1} or v_{i+1}, and therefore i and j have the same parity. Also, if a vertex v_i appears twice, then there are at most 3 elements of $D(i)$ with parity opposite of i. One of them will be the unique k such that $v_i=v_k$. For any other such index $j\neq k$, v_j must be the opposite parity occurrence of either v_{k-1} or v_{k+1}.

Case 1. *Some vertex v_i appears in the circuit exactly once and $s\geq 3$.* By relabeling, if necessary, we may assume that $i+3\leq 2s$. As noted above, any element of $D(i)$ has the same parity as i, so in particular $i+3\notin D(i)$. Thus we may apply Lemma 2.1 with indices i and $j=i+3$. The lemma says that we can reverse C by reversing two shorter circuits of length $(i+3)-i+1=4$ and $2s-(i+3)+i+1=2s-2$. By the induction hypothesis we can accomplish this with at most $(\frac{4}{2}-1)+(\frac{2s-2}{2}-1)=s-1$ switchings.

Case 2. *Each vertex appears exactly twice, once with each parity, and $s\geq 6$.* Since there are at most 3 elements of $D(1)$ with even parity, $\exists j\in\{4,6,8,10\}-D(1)$. Applying Lemma 2.1 with $i=1$ and j implies that we can reverse C by reversing two shorter circuits of length $j-1+1$ and $2s-(j-1)+1$. By the induction hypothesis we can accomplish this in at most $(\frac{j}{2}-1)+(\frac{2s-j+2}{2}-1)=s-1$ steps.

Remaining cases. Each vertex appears exactly twice, once with each parity, and $s\in\{3,4,5\}$. This means that the edges of the alternating circuit comprise two edge-disjoint 2-regular graphs on s vertices, one blue and the other red. For $s=3$ or 4, this is impossible. For $s=5$, the only possibility is shown in Figure 2.2, where we give an explicit sequence of length 3 to reverse C: first $\{ab,cd\}\to\{ac,bd\}$, then $\{ae,bc\}\to\{ab,ec\}$, and finally $\{ab,de\}\to\{ad,be\}$. ∎

Lemma 2.2 yields the following lemma, which, for $k=m(G\Delta H)$, provides half of our desired equality.

LEMMA 2.3. *If G and H have the same degrees and $G\Delta H$ can be partitioned into k symmetric circuits, then $d(G,H) \le \frac{1}{2}|E(G\Delta H)| - k$.*

Proof. By assumption, $G\Delta H$ can be partitioned into symmetric circuits $C_1,\ldots,C_k$ with lengths $2s_1,\ldots,2s_k$ so that $\sum_{i=1}^{k} 2s_i = |E(G\Delta H)|$. Reversing all of the corresponding alternating circuits C_i in G transforms G into H. By Lemma 2.2, reversing all of these alternating circuits requires at most $\sum_{i=1}^{k}(s_i - 1) = |E(G\Delta H)| - k$ switchings. ∎

The next result shows that a switching can decrease the quantity $\frac{1}{2}|E(G\Delta H)| - m(G\Delta H)$ by at most one. Since $G = H$ if and only if $\frac{1}{2}|E(G\Delta H)| - m(G\Delta H) = 0$, it follows that $d(G,H) \ge \frac{1}{2}|E(G\Delta H)| - m(G\Delta H)$.

LEMMA 2.4. *Let G and H have the same degrees and let G' be obtained from G by a switching. If $F = G\Delta H$ and $F' = G'\Delta H$, then*

$$\left(\frac{1}{2}|E(F)| - m(F)\right) - \left(\frac{1}{2}|E(F')| - m(F')\right) \le 1.$$

Proof. Since both $|E(F)|$ and $|E(F')|$ are even, we can choose t so that $2t = |E(F)| - |E(F')|$. Most of the work for this lemma is to establish the following claim: If F' can be partitioned into k symmetric circuits, then F can be partitioned into $k+t-1$ symmetric circuits. The claim guarantees that $m(F) \ge m(F') + t - 1$, which makes it easy to compute

$$\begin{aligned}\left(\frac{|E(F)|}{2} - m(F)\right) - \left(\frac{|E(F')|}{2} - m(F')\right) &= t - m(F) + m(F') \\ &\le t - (m(F') + t - 1) + m(F') = 1.\end{aligned}$$

Proof of claim. Let $\{ab, cd\} \to \{ac, bd\}$ be the switching in G and suppose F' can be partitioned into k symmetric circuits. The proof is broken into cases depending on which of the edges in $T = \{ab, bd, cd, ac\}$ are present in H. Associated with each case is a picture in Figure 2.3 showing edges of G and H on the left and edges of G' and H on the right with respect to these four vertex pairs. The solid lines represent edges of G or G'; the dashed, edges of H.

Case (i). Suppose $E(H) \cap T = \{ac, bd\}$. Here $t - 1 = 1$, so we must show that F can be partitioned into $k+1$ symmetric circuits. This is easy. Simply add the symmetric circuit (a, c, b, d, a) to the partition of F'.

Case (ii). Suppose $E(H) \cap T = \{ac\}$. Here $t-1 = 0$, so we must show that F can be partitioned into k symmetric circuits. Take the symmetric circuit in F' containing the edge bd and replace bd with the path $bacd$ to obtain a partition of F with the same number of circuits. By symmetry, this also covers the case $E(H) \cap T = \{bd\}$.

Case (iii). Suppose $E(H) \cap T = \{ac, bd, ab\}$. Similarly to Case (ii), replace the edge ab in the partition of F' with the path $acdb$ to obtain a partition of F. By symmetry this also covers the case $E(H) \cap T = \{ac, bd, cd\}$.

Case (iv). Suppose $E(H) \cap T$ is empty. Here $t - 1 = -1$, so we must show that F can be partitioned into at least $k-1$ symmetric circuits. First, suppose that in the partition of F' the edges ac and bd are in separate circuits, $C_1 = (a, c, u_1, \ldots, u_r a)$ and $C_2 = (d, b, v_1, \ldots, v_s, d)$. Then we can replace C_1 and C_2 with the single circuit $(a, b, v_1, \ldots, v_s, d, c, u_1, \ldots, u_r, a)$. Second, suppose ac and bd are in the same circuit C. Starting with the edge from a to c, if d occurs before b then replace $C = (a, c,\ u_1, \ldots, u_r,\ d, b, v_1, \ldots, v_s, a)$ with the two circuits $(d, c, u_1, \ldots, u_r, d)$ and

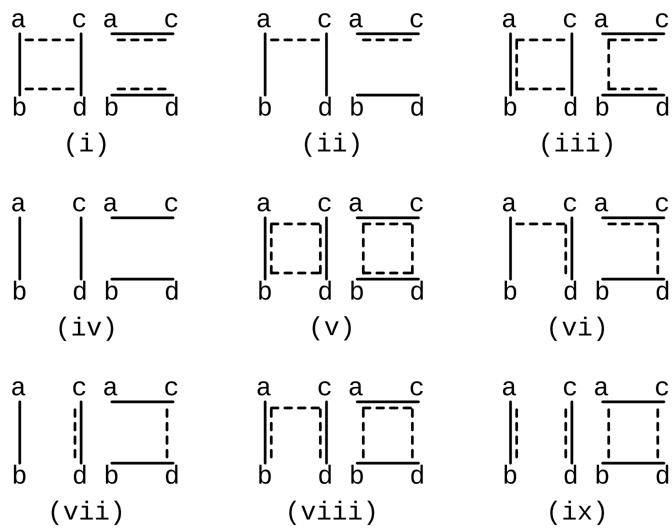


Fig. 2.3.

$(a, b, v_1, \ldots, v_s, a)$. Alternatively, if b occurs before d, then replace $C = (a, c, u_1, \ldots, u_r, b, d, v_1, \ldots, v_s, a)$ with the circuit $(a, b, u_r, \ldots, u_1, c, d, v_1, \ldots, v_s, a)$. For each possibility we have produced a partition of F containing at least $k-1$ symmetric circuits.

Case (v). Suppose $E(H) \cap T = T$. Again $t - 1 = -1$ and we can apply the same argument as in Case (iv).

We now pause to make a general argument which we will apply in the remaining cases. Let $S = v_1, v_2, \ldots, v_r$ be a trail in F' whose edges alternate between $E(G') - E(H)$ and $E(H) - E(G')$. Let $C_1, \ldots, C_r$ be the symmetric circuits in the partition of F' which contain any edge from S. Then, by the argument in Lemma 1.1, we can repartition the edges in $C_1, \ldots, C_r$ into possibly different symmetric circuits $C'_1, \ldots, C'_s$ where S occurs as a subtrail of circuit C'_1.

Case (vi). Suppose $|E(H) \cap \{ac, bd\}| = |E(H) \cap \{ab, cd\}| = 1$. By symmetry we may assume $E(H) \cap T = \{ac, cd\}$. Here $t - 1 = -1$, so we must show how F can be partitioned into at least $k - 1$ symmetric circuits. Let $C_1, \ldots, C_r$ be the symmetric circuits in the partition of F' which contain either bd or cd. Repartition the edges in $C_1, \ldots, C_r$ into symmetric circuits $C'_1, \ldots, C'_s$ where the path bdc occurs as a subtrail of circuit C'_1. Form a partition of F by replacing $C_1, \ldots, C_r$ with $C'_1, \ldots, C'_s$ and replacing the path bdc in C'_1 with the path bac. Since $r \leq 2$ and $s \geq 1$ the partition contains at least $k - 1$ symmetric circuits.

Case (vii). Suppose $E(H) \cap T = \{cd\}$. Here $t-1 = -2$, so we must show how F can be partitioned into at least $k-2$ symmetric circuits. Let $C_1, \ldots, C_r$ be the symmetric circuits in the partition of F' which contain any of $\{ac, cd, db\}$. Repartition the edges in $C_1, \ldots, C_r$ into symmetric circuits $C'_1, \ldots, C'_s$ where the path $acdb$ occurs as a subtrail of circuit C'_1. Form a partition of F by replacing $C_1, \ldots, C_r$ with $C'_1, \ldots, C'_s$ and replacing the path $acdb$ in C'_1 with the edge ab. Since $r \leq 3$ and $s \geq 1$, the partition contains at least $k-2$ symmetric circuits. By symmetry, this argument also covers the case $E(H) \cap T = \{ab\}$.

Case (viii). Suppose $E(H) \cap T = \{ba, ac, cd\}$. Here $t - 1 = -2$, so we must show

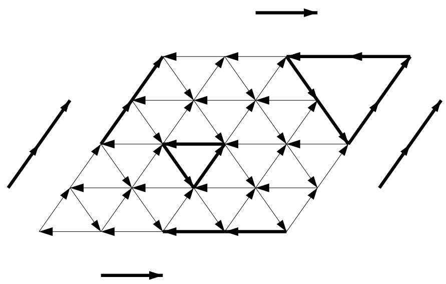


FIG. 3.1. $L_{i,j}$ *corresponding to literal* $l_{i,j}$.

how F can be partitioned into at least $k-2$ symmetric circuits. The argument is the same as in Case (vii) and by symmetry can also be applied to the case $E(H) \cap T = \{ba, bd, cd\}$.

Case (ix). Suppose $E(H) \cap T = \{ab, cd\}$. Here $t-1 = -3$, so we must show how F can be partitioned into at least $k-3$ symmetric circuits. Let $C_1, \ldots, C_r$ be the symmetric circuits in the partition of F' which contain any edge of C. Repartition the edges in $C_1, \ldots, C_r$ into symmetric circuits $C'_1, \ldots, C'_s$ where $C'_1 = acdba$. Form a partition of F by replacing $C_1, \ldots, C_r$ with $C'_2, \ldots, C'_s$. Since $r \leq 4$ and $s \geq 1$ we have at least $k-3$ circuits, unless $s = 1$. But in that case r must be equal to 1 as well; thus we have $k-1$ circuits. ∎

Lemma 2.3 and Lemma 2.4 together establish the following theorem.

THEOREM 2.5. *For graphs* G *and* H *with the same degrees,* $d(G, H) = \frac{1}{2}|E(G\Delta H)| - m(G\Delta H)$.

3. Computational complexity. In this section, we demonstrate that computing $m(G\Delta H)$, or equivalently, $d(G, H)$, is NP-complete. To do this we introduce two decision problems.

SYMMETRIC CIRCUITS PARTITION (SCP): *Given two simple graphs* G, H *with the same degrees and an integer* k*, is there an edge-partition of* $G\Delta H$ *into* k *or more symmetric circuits?*

DISJOINT DIRECTED TRIANGLES (DDT): *Given an oriented graph with equal in-degree and out-degree at each vertex, is there a partition of the arcs into directed* 3*-cycles?*

First, by modifying a proof of Holyer [2], we reduce 3-SAT, a well-known NP-complete problem, to DDT. Afterwards, using a much simpler argument, we reduce DDT to SCP.

THEOREM 3.1. DISJOINT DIRECTED TRIANGLES *(DDT) is NP-hard.*

Proof. Let $C_1, \ldots, C_r$ with variables $\{x_1, \ldots, x_s\}$ be the clauses in an instance of 3-SAT, where each clause C_i consists of three literals $\{l_{i,1}, l_{i,2}, l_{i,3}\}$. Our goal is to construct an oriented graph G that can be partitioned into directed 3-cycles if and only if the given instance of 3-SAT is satisfiable.

For each literal $l_{i,j}$ we create a copy $L_{i,j}$ of the toroidal digraph in Figure 3.1, with the arrows indicating the edges to be identified (each edge on the outside boundary of the graph appears twice). In each of the figures, bold arcs will appear in exactly three 3-cycles of G while light arcs will appear in exactly two 3-cycles. Let $T_{i,j}$ be the 3-cycle of bold edges in $L_{i,j}$.

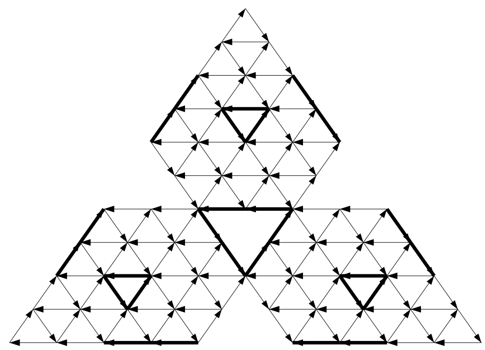


FIG. 3.2. H_i *corresponding to clause* C_i.

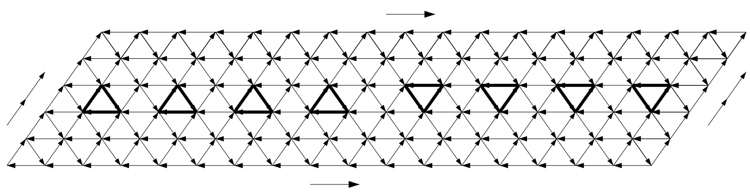


FIG. 3.3. G_x *corresponding to variable* x.

For each clause $C_i = \{l_{i,1}, l_{i,2}, l_{i,3}\}$, create a digraph H_i by identifying the graphs $L_{i,1}, L_{i,2}, L_{i,3}$ along the bold 6-cycles as shown in Figure 3.2. Since each $L_{i,j}$ is toroidal, each edge in the bold 6-cycle appears in exactly three 3-cycles of H_i.

For each variable x, create a toroidal graph G_x like the one shown in Figure 3.3. We refer to the bold 3-cycles which point up as *T-cycles*, those pointing down as *F-cycles*. Extend the graph to the left and right far enough so that there are as many T-cycles as there are occurrences of x as a literal, and as many F-cycles as there are occurrences of $\overline{x}$ as a literal.

Finally, to construct G, make the following identifications of 3-cycles. If $l_{i,j} = x$, then identify $T_{i,j}$ with a unique T-cycle of G_x. If $l_{i,j} = \overline{x}$, then identify $T_{i,j}$ with a unique F-cycle of G_x. This identification causes each edge of every bold 3-cycle to appear in exactly three 3-cycles.

Now suppose there is a partition P of G into directed triangles. Consider an edge e from the bold 6-cycle in some subgraph H_i. The partition must contain exactly one of the three 3-cycles in H_i containing e. Place a 1 in this 3-cycle to indicate that it is in the partition and then apply the following inference rules:

- Place a 0 in a 3-cycle if it shares any arc with a 3-cycle containing a 1.
- Place a 1 in a 3-cycle if it shares a light arc with a 3-cycle containing a 0.

Placing a 0 in a 3-cycle indicates that the 3-cycle is not in the partition. Thus the first inference rule is justified since P is a partition of the edges. The second rule is justified since each light arc appears in only two 3-cycles. After applying the two rules as often as possible, place an * in any remaining 3-cycles.

This procedure leads to three possible partitions of H_i, depending on which of the three 3-cycles using e appears in the partition. In each case, one of the bold 3-cycles contains a 0 and the other two bold 3-cycles contain *'s. Moreover, in each case there is a 1 in each of the 3-cycles in H_i that shares an arc with the bold 3-cycle containing 0.

Depending on the partition P, applying the same inference rules to each subgraph G_x results in either all of the F-cycles containing * and all the T-cycles containing 0 or vice versa. Moreover, in each case there is a 1 in each of the 3-cycles in G_x that shares an arc with a bold 3-cycle containing 0. Taken together with our assessment of H_i, we learn that the bold 3-cycle in H_i containing 0 cannot be identified with a bold 3-cycle in G_x containing 0 since this would result in the edges of the identified 3-cycle appearing in more than one 3-cycle of P.

We now make a satisfying assignment. If the T-cycles of G_x contain *'s, then let x be true. On the other hand, if the F-cycles of G_x contain *'s, then let x be false. We claim that this assignment satisfies each C_i. To show this, we need to show for each i that at least one of $\{l_{i,1}, l_{i,2}, l_{i,3}\}$ is true. Let $T_{i,j}$ be the bold 3-cycle of H_i containing the 0.

Case (1). If $l_{i,j} = x$, then $T_{i,j}$ is identified with a T-cycle of G_x. We already have argued that $T_{i,j}$ cannot be identified with a 3-cycle containing a 0, so the T-cycle must contain a *, which implies that x is true and so satisfies the clause.

Case (2). If $l_{i,j} = \overline{x}$, then $T_{i,j}$ is identified with an F-cycle of G_x. Again we know that $T_{i,j}$ cannot be identified with a 3-cycle containing a 0, so the F-cycle must contain a * which implies that x is false and so $\overline{x}$ satisfies the clause.

In the other direction, suppose there exists a satisfying assignment. If the variable x is assigned true, then place 0's in the F-cycles of G_x and apply the inference rules. If the variable x is assigned false, then place 0's in the T-cycles of G_x and apply the inference rules. For each clause C_i, choose one true literal $l_{i,j}$ and partition H_i so that $T_{i,j}$ is the bold 3-cycle containing the 0. Let P consist of all 3-cycles containing a 1. The only edges of G not in P are those in 3-cycles formed by the identification of two *'s. These can be added to P to complete the partition. ∎

We now show formally that computing $m(G\Delta H)$ is NP-complete by showing that SCP is NP-complete.

THEOREM 3.2. *SCP is NP-complete.*

Proof. It is easy to determine whether k or more circuits are all symmetric and partition $G\Delta H$. Hence SCP is in NP. To see that SCP is NP-hard, we reduce from DDT. An instance of DDT consists of a digraph D with equal in-degree and out-degree at each vertex. Form a two-colored simple graph from D by subdividing each arc $x \to y$ into $x \to m_{xy} \to y$ and replacing $x \to m_{xy}$ with a red edge and $m_{xy} \to y$ with a blue edge. Let G be the subgraph of red edges, and let H be the subgraph of blue edges. Note that G and H have the same vertex degrees and that the shortest circuit in $G\Delta H$ has length 6. If D has $3t$ arcs, then $G\Delta H$ has $6t$ edges and can be edge-partitioned into at least t symmetric circuits if and only if D can be partitioned into directed triangles. ∎

REFERENCES

[1] R. B. EGGLETON AND D. A. HOLTON, *Graphic sequences*, in Combinatorial Mathematics VI, Proceedings of the Sixth Australian Conference, University of New England, Armidale, 1978, Springer, Berlin, 1979, pp. 1–10.

[2] I. HOLYER, *The NP-completeness of some edge-partition problems*, SIAM J. Comput., 10 (1981), pp. 713–717.

[3] Z. Majcher, *Alternating cycles and realizations of a degree sequence*, Comment. Math. Univ. Carolin., 28 (1987), pp. 467–480.
[4] R. Taylor, *Constrained switchings in graphs*, in Combinatorial Mathematics VIII (Geelong, 1980), Springer, Berlin, 1981, pp. 314–336.
[5] R. Taylor, *Switchings constrained to* 2*-connectivity in simple graphs*, SIAM J. Alg. Discrete Methods, 3 (1982), pp. 114–121.
[6] D. B. West, *Introduction to Graph Theory*, Prentice-Hall, Upper Saddle River, NJ, 1996.

SIAM J. DISCRETE MATH.
Vol. 12, No. 3, pp. 307–316

EDGE-BANDWIDTH OF GRAPHS*

TAO JIANG†, DHRUV MUBAYI‡, ADITYA SHASTRI§, AND DOUGLAS B. WEST†

Abstract. The *edge-bandwidth* of a graph is the minimum, over all labelings of the edges with distinct integers, of the maximum difference between labels of two incident edges. We prove that edge-bandwidth is at least as large as bandwidth for every graph, with equality for certain caterpillars. We obtain sharp or nearly sharp bounds on the change in edge-bandwidth under addition, subdivision, or contraction of edges. We compute edge-bandwidth for K_n, $K_{n,n}$, caterpillars, and some theta graphs.

Key words. bandwidth, edge-bandwidth, clique, biclique, caterpillar

AMS subject classifications. 05C78, 05C35

PII. S0895480197330758

1. Introduction.

A classical optimization problem is to label the vertices of a graph with distinct integers so that the maximum difference between labels on adjacent vertices is minimized. For a graph G, the optimal bound on the differences is the *bandwidth* $B(G)$. The name arises from computations with sparse symmetric matrices, where operations run faster when the matrix is permuted so that all entries lie near the diagonal. The bandwidth of a matrix M is the bandwidth of the corresponding graph whose adjacency matrix has a 1 in those positions where M is nonzero. Early results on bandwidth are surveyed in [2] and [3].

In this paper, we introduce an analogous parameter for edge-labelings. An *edge-numbering* (or *edge-labeling*) of a graph G is a function f that assigns distinct integers to the edges of G. We let $B'(f)$ denote the maximum of the difference between labels assigned to adjacent (incident) edges. The *edge-bandwidth* $B'(G)$ is the minimum of $B'(f)$ over all edge-labelings. The term "edge-numbering" is used because we may assume that f is a bijection from $E(G)$ to the first $|E(G)|$ natural numbers.

We use the notation $B'(G)$ for the edge-bandwidth of G because it is immediate that the edge-bandwidth of a graph equals the bandwidth of its line graph. Thus well-known elementary bounds on bandwidth can be applied to line graphs to obtain bounds on edge-bandwidth. We mention several such bounds. We compute edge-bandwidth on a special class where all these bounds are arbitrarily bad.

The relationship between edge-bandwidth and bandwidth is particularly interesting. Always $B(G) \leq B'(G)$, with equality for caterpillars of diameter more than k in which every vertex has degree 1 or $k+1$. Among forests, $B'(G) \leq 2B(G)$, which is almost sharp for stars. More generally, if G is a union of t forests, then $B'(G) \leq 2tB(G) + t - 1$.

Chvátalová and Opatrný [5] studied the effect on bandwidth of edge addition, contraction, and subdivision (see [22] for further results on edge addition). We study

*Received by the editors December 1, 1997; accepted for publication (in revised form) January 8, 1999; published electronically September 7, 1999.

http://www.siam.org/journals/sidma/12-3/33075.html

†Department of Mathematics, University of Illinois, Urbana, IL 61801-2975 (west@math.uiuc.edu, j-tao@math.uiuc.edu).

‡School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332-0160 (mubayi@math.gatech.edu).

§Department of Computer Science, Banasthali University, P.O. Banasthali Vidyapith, Rajasthan 304 022, India (shastri@bv.ernet.in).

these for edge-bandwidth. Adding or contracting an edge at most doubles the edge-bandwidth. Subdividing an edge decreases the edge-bandwidth by at most a factor of 1/3. All these bounds are sharp within additive constants. Surprisingly, subdivision can also increase edge-bandwidth, but at most by 1, and contraction can decrease it by 1.

Because the edge-bandwidth problem is a restriction of the bandwidth problem, it may be easier computationally. Computation of bandwidth is NP-complete [17], remaining so for trees with maximum degree 4 [8] and for several classes of caterpillar-like graphs [11, 16]. Such graphs generally are not line graphs (they contain claws). It remains open whether computing edge-bandwidth (computing bandwidth of line graphs) is NP-hard.

Due to the computational difficulty, bandwidth has been studied on various special classes. Bandwidth has been determined for caterpillars and for various generalizations of caterpillars [1, 11, 14, 21], for complete k-ary trees [19], for rectangular and triangular grids [4, 10] (higher dimensions [9, 15]), for unions of pairwise internally disjoint paths with common endpoints (called "theta graphs" [6, 13, 18]), etc. Polynomial-time algorithms exist for computing bandwidth for graphs in these classes and for interval graphs [12, 20]. We begin analogous investigations for edge-bandwidth by computing the edge-bandwidth for cliques, for equipartite complete bipartite graphs, and for some theta graphs.

2. Relation to other parameters.

We begin by listing elementary lower bounds on edge-bandwidth that follow from standard arguments about bandwidth when applied to line graphs.

PROPOSITION 2.1. *Edge-bandwidth satisfies the following:*
(a) $B'(H) \le B'(G)$ *when* H *is a subgraph of* G.
(b) $B'(G) = \max\{B'(G_i)\}$, *where* $\{G_i\}$ *are the components of* G.
(c) $B'(G) \ge \Delta(G) - 1$.

Proof. (a) A labeling of G contains a labeling of H. (b) Concatenating labelings of the components achieves the lower bound established by (a). (c) The edges incident to a single vertex induce a clique in the line graph. The lowest and highest among these labels are at least $\Delta(G) - 1$ apart. ∎

PROPOSITION 2.2. $B'(G) \ge \max_{H \subseteq G} \left\lceil \frac{e(H)-1}{\text{diam }(L(H))} \right\rceil$.

Proof. This is the statement of Chung's "density bound" [3] for line graphs. Every labeling of a graph contains a labeling of every subgraph. In a subgraph H, the lowest and highest labels are at least $e(H) - 1$ apart, and the edges receiving these labels are connected by a path of length at most diam $(L(H))$, so by the pigeonhole principle some consecutive pair of edges along the path have labels differing by at least $(e(H) - 1)/\text{diam }(L(H))$. ∎

Subgraphs of diameter 2 include stars, and a star in a line graph is generated from an edge of G with its incident edges at both endpoints. The size of such a subgraph is at most $d(u) + d(v) - 1$, yielding the bound $B'(G) \ge [d(u) + d(v)]/2 - 1$ for $uv \in E(G)$. This is at most $\Delta(G) - 1$, the lower bound from Proposition 2.1. Nevertheless, because of the way in which stars in line graphs arise, they can yield a better lower bound for regular or nearly regular graphs. We develop this next.

PROPOSITION 2.3. *For* $F \subseteq E(G)$, *let* $\partial(F)$ *denote the set of edges not in* F *that are incident to at least one edge in* F. *The edge-bandwidth satisfies* $B'(G) \ge \max_k \min_{|F|=k} |\partial(F)|$.

Proof. This is the statement of Harper's "boundary bound" [9] for line graphs.

Some set F of k edges must be the set given the k smallest labels. If m edges outside this set have incidents with this set, then the largest label on the edges of ∂F is at least $k+m$, and the difference between the labels on this and its incident edge in F is at least m. □

COROLLARY 2.4. $B'(G) \geq \min_{uv \in E(G)} d(u) + d(v) - 2$.

Proof. We apply Proposition 2.3 with $k = 1$. Each edge uv is incident to $d(u) + d(v) - 2$ other edges. Some edge must have the least label, and this establishes the lower bound. □

Although these bounds are often useful, they can be arbitrarily bad. The *theta graph* $\Theta(l_1, \ldots, l_m)$ is the graph that is the union of m pairwise internally disjoint paths with common endpoints and lengths $l_1, \ldots, l_m$. The name "theta graph" comes from the case $m = 3$. The bandwidth is known for all theta graphs, but settling this was a difficult process finished in [18]. When the path lengths are equal, the edge-bandwidth and bandwidth both equal m, using the density lower bound and a simple construction. The edge-bandwidth can be much higher when the lengths are unequal. Our example showing this will later demonstrate sharpness of some bounds.

Our original proof of the lower bound was lengthy. The simple argument presented here originated with Dennis Eichhorn and Kevin O'Bryant. It will be generalized in [7] to compute edge-bandwidth for a large class of theta graphs.

Example A. Consider $G = \Theta(l_1, \ldots, l_m)$ with $l_m = 1$ and $l_1 = \cdots = l_{m-1} = 3$. Let a_i, b_i, c_i denote the edges of the ith path of length 3, and let e be the edge incident to all a_i's at one end and to all c_i's at the other end. Since $\Delta(G) = m$, Proposition 2.1(c) yields $B'(G) \geq m-1$. Proposition 2.2 also yields $B'(G) \geq m-1$. For $1 \leq k \leq 2m-2$, the first k edges in the list $a_1, \ldots, a_{m-1}, b_1, \ldots, b_{m-1}$ are together incident to exactly m other edges, and larger sets are incident to at most $m-1$ other edges. Thus the best lower bound from Proposition 2.3 is at most m.

Nevertheless, $B'(G) = \lceil (3m-3)/2 \rceil$. For the upper bound, we assign the $3m-2$ labels in order to a's, b's, and c's, inserting e before $b_{\lceil m/2 \rceil}$. The difference between labels of incidence edges is always at most m except for incidences involving e, which are at most $\lceil (3m-3)/2 \rceil$ since e has the middle label.

$$a_1, \ldots, a_{m-1}, b_1, \ldots, b_{\lceil m/2 \rceil - 1}, e, b_{\lceil m/2 \rceil}, \ldots, b_{m-1}, c_1, \ldots, c_{m-1}$$

To prove the lower bound, consider a numbering f of $E(G)$ by distinct integers and let $k = B'(f)$. Let $\alpha = \max\{f(e), \max_i\{f(a_i)\}\}$ and $\alpha' = \min\{f(e), \min_i\{f(c_i)\}\}$. Comparing the edges with labels $\alpha, f(e), \alpha'$ yields $\alpha - k \leq f(e) \leq \alpha' + k$. Let I be the interval $[\alpha - k, \alpha' + k]$. By construction, I contains the labels of all a's, all c's, and e. If $f(a_i) < \alpha'$ and $f(c_i) > \alpha$, then also $f(b_i) \in I$. By the choice of α, α', avoiding this requires $\alpha' < f(a_i) \leq \alpha$ or $\alpha' \leq f(c_i) < \alpha$. Since each label is assigned only once and the label $f(e)$ cannot play this role, only $\alpha - \alpha'$ of the b's can have labels outside I. Counting the labels we have forced into I yields $|I| \geq (2m-1) + (m-1-\alpha+\alpha')$. On the other hand, $|I| = 2k + \alpha' - \alpha + 1$. Thus $k \geq (3m-3)/2$, as desired. □

3. Edge-bandwidth vs. bandwidth. In this section we prove various best-possible inequalities involving bandwidth and edge-bandwidth. The proof that $B(G) \leq B'(G)$ requires several steps. All steps are constructive. When f or g is a labeling of the edges or vertices of G, we say that $f(e)$ of $g(v)$ is the *f-label* or *g-label* of the edge e or vertex v. An *f-label* on an edge incident to u is an *incident f-label* of u.

LEMMA 3.1. *If a finite graph G has minimum degree at least two, then $B(G) \leq B'(G)$.*

Proof. From an optimal edge-numbering f (such that $B'(f) = B'(G) = m$), we define a labeling g of the vertices. The labels used by g need not be consecutive, but we show that $|g(u) - g(v)| \leq m$ when u and v are adjacent.

We produce g in phases. At the beginning of each phase, we choose an arbitrary unlabeled vertex u and call it the *active vertex*. At each step in a phase, we select the unused edge e of smallest f-label among those incident to the active vertex. We let $f(e)$ be the g-label of the active vertex, mark e *used*, and designate the other endpoint of e as the active vertex. If the new active vertex already has a label, we end the phase. Otherwise, we continue the phase.

When we examine a new active vertex, it has an edge with least incident label, because every vertex has degree at least 2 and we have not previously reached this vertex. Each phase eventually ends, because the vertex set is finite and we cannot continue reaching new vertices. The procedure assigns a label $g(u)$ for each $u \in V(G)$, since we continue to a new phase as long as an unlabeled vertex remains.

It remains to verify that $|g(u) - g(v)| \leq m$ when $uv \in E(G)$. Suppose that $g(u) = a = f(e)$ and $g(v) = b = f(e')$. Since each vertex is assigned the f-label of an incident edge, we have e, e' incident to u, v, respectively. If the edge uv is one of e, e', then e and e' are incident, which implies that $|g(u) - g(v)| = |f(e) - f(e')| \leq m$.

Otherwise, we have $f(uv) = c$ for some other value c. We may assume that $a < b$ by symmetry. If $a < c$ and $b < c$, then $|g(u)-g(v)| = b-a < c-a = f(uv)-f(e) \leq m$. Thus we may assume that $b > c$. In particular, $g(v)$ is not the least f-label incident to v.

The algorithm assigns v a label when v first becomes active, using the least f-label among *unused* incident edges. When v first becomes active, only the edge of arrival is a used incident edge. Thus $g(v)$ is the least incident f-label except when v is first reached via the least-labeled incident edge. In this case, $g(v)$ is the second smallest incident f-label. Thus c is the least f-label incident to v and v becomes active by arrival from u. This requires $g(u) = c$, which contradicts $g(u) = a$ and eliminates the bad case. ∎

LEMMA 3.2. *If G is a tree, then $B(G) \leq B'(G)$.*

Proof. Again we use an optimal edge-numbering f to define a vertex-labeling g whose adjacent vertices differ by at most $B'(f)$. We may assume that the least f-label is 1, occurring on the edge $e = uv$. Assign (temporarily) $g(u) = g(v) = f(e)$. View the edge e as the root of G. For each vertex $x \notin \{u, v\}$, let $g(x)$ be the f-label of the edge incident to x along the path from x to the root.

If $xy \in E(G)$ and $xy \neq uv$, then we may assume that y is on the path from x to the root. We have assigned $g(x) = f(xy)$, and $g(y)$ is the f-label of an edge incident to y, so $|g(x) - g(y)| \leq B'(f)$.

Our labeling g fails to be the desired labeling only because we used 1 on both u and v. Observe that the largest f-label incident to uv occurs on an edge incident to u or on an edge incident to v but not both; we may assume the latter. Now we change $g(u)$ to 0. Because the differences between $f(uv)$ and f-labels on edges incident to u were *less* than $B'(f)$, this produces the desired labeling g. ∎

THEOREM 3.3. *For every graph G, $B(G) \leq B'(G)$.*

Proof. By Proposition 2.1(b), it suffices to consider connected graphs. Let f be an optimal edge-numbering of G; we produce a vertex labeling g. Lemma 3.2 applies when G is a tree. Otherwise, G contains a cycle, and iteratively deleting vertices of degree 1 produces a subgraph G' in which every vertex has degree at least 2. The algorithm of Lemma 3.1, applied to the restriction of f to G', produces a vertex labeling g of G' in which (1) adjacent vertices have labels differing by at most $B'(f)$, and (2) the label on each vertex is the f-label of some edge incident to it in G'.

To obtain a vertex labeling of G, reverse the deletion procedure. This iteratively adds a vertex x adjacent to a vertex y that already has a g-label. Assign to x the f-label of the edge xy in the full edge-numbering f of G. Now $g(x)$ and $g(y)$ are the f-labels of two edges incident to y in G, and thus $|g(x) - g(y)| \leq B'(f)$. The claims (1) and (2) are preserved, and we continue this process until we replace all vertices that were deleted from G. □

A *caterpillar* is a tree in which the subtree obtained by deleting all leaves is a path. One of the characterizations of caterpillars is the existence of a linear ordering of the edges such that each prefix and each suffix forms a subtree. We show that such an ordering is optimal for edge-bandwidth and use this to show that Theorem 3.3 is nearly sharp.

PROPOSITION 3.4. *If G is a caterpillar, then $B'(G) = \Delta(G) - 1$. Let G be the caterpillar of diameter d in which every vertex has degree $k+1$ or 1. If $d \geq k$, then $B(G) = B'(G) = k$.*

Proof. Let G be a caterpillar. Let $v_1, \ldots, v_{d-1}$ be the nonleaf vertices of the dominating path. The diameter of G is d. Number the edges by assigning labels in the following order: first the pendant edges incident to v_1, then v_1v_2, then the pendant edges incident to v_2, then v_2v_3, etc. Since edges are incident only at $v_1, \ldots, v_{d-1}$, this ordering places all pairs of incident edges within $\Delta(G) - 1$ positions of each other. Since $B'(G) \geq \Delta(G) - 1$ for all G, equality holds.

For a caterpillar G with order n and diameter d, Chung's density bound yields $B(G) \geq (n-1)/d$. Let G be the caterpillar of diameter d in which every vertex has degree $k+1$ or 1. We have $d-1$ vertices of degree $k+1$, so $n = (d-1)k + 2$ and $B(G) > k - k/d$. When $d \geq k$, we have $B(G) \geq k$.

On the other hand, we have observed that $B'(G) \leq \Delta(G) - 1 = k$ for caterpillars. By Theorem 3.3, equality holds throughout for these special caterpillars. □

Theorem 3.3 places a lower bound on $B'(G)$ in terms of $B(G)$. We next establish an upper bound. The *arboricity* is the minimum number of forests needed to partition the edges of G.

THEOREM 3.5. *If G has arboricity t, then $B'(G) \leq 2tB(G) + t - 1$. When $t = 1$, the inequality is almost sharp; there are caterpillars with $B'(G) = 2B(G) - 1$.*

Proof. Given an optimal number g of $V(G)$, we construct a labeling f of $E(G)$. Let $G_1, \ldots, G_t$ be a decomposition of G into the minimum number of forests. In each component of each G_i, select a root. Each edge of G_i is the first edge on the path from one of its endpoints to the root of its component in G_i; for $e \in E(G_i)$, let $v(e)$ denote this endpoint. Define $f(e) = tg(v(e)) + i$.

Each vertex of each forest heads toward the root of its component in that forest along exactly one edge, so the f-labels of the edges are distinct. Each f-label arises from the g-label of one of its endpoints. Thus the f-labels of two incident edges arise from the g-labels of vertices separated by distance at most 2 in G. Also, the indices of the forests containing these edges differ by at most $t-1$. Thus when e, e' are incident we have $|f(e) - f(e')| \leq t2B(g) + t - 1$.

The bandwidth of a caterpillar is the maximum density (number of edges divided by diameter) over subtrees [14]. This equals $\lceil \Delta(G)/2 \rceil$ whenever the vertex degrees all lie in $\{\Delta(G), 2, 1\}$ and the vertices of degree $\Delta(G)$ are pairwise. (Without [14], this still holds explicitly for stars.) □

4. Effect of edge operations. In this section, we obtain bounds on the effect of local edge operations on the edge-bandwidth. The variations can be linear in the value of the edge-bandwidth, and our bounds are optimal except for additive constants. We study addition, subdivision, and contraction of edges.

THEOREM 4.1. *If H is obtained from G by adding an edge, then $B'(G) \le B'(H) \le 2B'(G)$. Furthermore, for odd k there are examples of $H = G+e$ such that $B'(G) = k$ and $B'(H) \ge 2k-1$.*

Proof. The first inequality holds because G is a subgraph of H. For the second, let g be an optimal edge-numbering of G; we produce an edge-numbering f of H such that $B'(f) \le 2B'(g)$.

If e is not incident to an edge of G, form f from g by giving e a new label higher than the others. If only one endpoint of e is incident to an edge e' of G, form f by leaving the g-labels less than $g(e')$ unchanged, augmenting the remaining labels by 1, and letting $f(e) = g(e') + 1$. We have $B(f) \le B(g) + 1$.

Thus we may assume that the new edge e joins two vertices of G. Our construction for this case modifies an argument in [22]. Let e_i be the edge such that $g(e_i) = i$, for $1 \le i \le B(g)$. Let p, q be the smallest and largest indices of edges of G incident to e, respectively, and let $r = \lfloor (p+q)/2 \rfloor$.

The idea in defining f from g is to "fold" the ordering at r, renumbering out from there so that e_p and e_q receive consecutive labels, and inserting e just before this. The renumbering of the old edges is as follows:

$$f(e_j) = \begin{cases} 2(j-r) & \text{if } r < j < q, \\ 2(j-r)+1 & \text{if } q \le j, \\ 2(r-j)+1 & \text{if } p < j \le r, \\ 2(r-j)+2 & \text{if } j \le p. \end{cases}$$

Finally, let $f(e) = \min\{f(e_p), f(e_q)\} - 1 = q - p$. After the edges with g-labels higher than q or lower than p are exhausted, the new numbering leaves gaps. For edges $e_i, e_j \in E(G)$, we have $|f(e_i) - f(e_j)| \le 2|i-j| + 1$, where the possible added 1 stems from the insertion of e. When r is between i and j, the actual stretch is smaller.

It remains to consider incidences involving e. Suppose that $e' = e_j$ is incident to e. Note that $1 \le f(e') \le q - p + 2 = f(e) + 2$; we may assume that $1 \le f(e') < f(e)$. If e_p and e_q are incident to the same endpoint of e, then $1 \le f(e) - f(e') \le q - p + 1 \le B(g) + 1$. If e_p and e_q are incident to opposite endpoints of e, then e' is incident to e_p or e_q. In these two cases, we have $p \le j \le p + B(g)$ or $q - B(g) \le j \le q$. Since j differs from p or q, respectively, by at most $B(g)$, we obtain $1 \le f(e) - f(e') \le 2B(g)$.

The bound is nearly sharp when k is odd. Let G be the caterpillar of diameter $k+1$ with vertices of degrees $k+1$ and 1 (see Proposition 3.4). We have $e(G) = k^2 + 1$ and $B'(G) = B(G) = k$. The graph H formed by adding the edge $v_1 v_k$ is a cycle of length k plus pendant edges; each vertex of the cycle has degree $k+1$ except for two adjacent vertices of degree $k+2$. The diameter of $L(H)$ is $\lfloor k/2 \rfloor + 1 = (k+1)/2$, and H has $k^2 + 2$ edges. By Proposition 2.2, we obtain $B'(H) \ge \left\lceil \frac{k^2+1}{(k+1)/2} \right\rceil = \left\lceil 2k - 2 + \frac{4}{k+1} \right\rceil = 2k - 1$. ☐

Subdividing an edge uv means replacing uv by a path u, w, v passing through a new vertex w. If H is obtained from G by subdividing one edge of G, then H is an *elementary subdivision* of G. Edge subdivision can reduce the edge-bandwidth considerably, but it increases the edge-bandwidth by at most one.

THEOREM 4.2. *If H is an elementary subdivision of G, then $\lceil (2B'(G) + \delta)/3 \rceil \le B'(H) \le B'(G) + 1$, where δ is 1 if $B'(H)$ is odd and 0 if $B'(H)$ is even, and these* *bounds are sharp.*

Proof. Suppose that H is obtained from G by subdividing edge e. From an optimal edge-numbering g of G, we obtain an edge-numbering of H by augmenting the labels greater than $g(e)$ and letting the labels of the two new edges be $g(e)$ and $g(e)+1$. This stretches the difference between incident labels by at most 1.

For sharpness of the bound, compare $G = \Theta(1, 2, \ldots, 2)$ and $G' = \Theta(1, 3, \ldots, 3)$, where each has m paths with common endpoints. In Example A, we proved that $B'(G') = \lceil 3(m-1)/2 \rceil$. In G, let the ith path have edges a_i, b_i for $i < m$, with e the extra edge. The ordering $a_1, \ldots, a_{m-1}, e, b_1, \ldots, b_{m-1}$ yields $B'(G) \le m$. The graph G' is obtained from G by a sequence of $m-1$ elementary subdivisions, roughly half of which must increase the edge-bandwidth. The desired graph H is the first where the bandwidth is $m+1$.

To prove the lower bound on $B'(H)$, we consider an optimal edge-numbering f of H and obtain an edge-numbering of G. For the edges e', e'' introduced to form H after deleting e, let $p = f(e')$ and $q = f(e'')$. We may assume that $p < q$. Let $r = \lfloor (p+q)/2 \rfloor$. Define g by leaving the f-labels below p and in $[r+1, q-1]$ unchanged, decreasing those in $[p+1, r]$ and above q by 1, and setting $g(e) = r$. The differences between labels on edges belonging to both G and H change by at most 1 and increase only when the difference is less than $B'(f)$. For incidents involving e, the incident edge ϵ was incident in H to e' or e''. The difference $|g(e) - g(\epsilon)|$ exceeds $B'(f)$ only if $g(\epsilon) < p$ or $g(\epsilon) > q$. In the first case, the difference increases by $r - p = \lfloor (q-p)/2 \rfloor$. In the second, it increases by $q - r - 1 = \lceil (q-p)/2 \rceil - 1$. We obtain $B'(G) \le B'(H) + \left\lfloor \frac{q-p}{2} \right\rfloor \le \left\lfloor \frac{3B'(H)}{2} \right\rfloor$. Whether $B'(H)$ is even or odd, this establishes the bound claimed.

For sharpness of the bound, compare $G = \Theta(1, 3, \ldots, 3)$ and $H = \Theta(2, 3, \ldots, 3)$. In H let the ith path have edges a_i, b_i, c_i for $i < m$, with d, e the remaining path. The ordering $a_1, \ldots, a_{m-1}, d, b_1, \ldots, b_{m-1}, e, c_1, \ldots, c_{m-1}$ yields $B'(H) \le m$. From Example A, $B'(G) = \lceil 3(m-1)/2 \rceil$. Whether m is odd or even, this example achieves the lower bound on $B'(H)$. □

Contracting an edge uv means deleting the edge and replacing its endpoints by a single combined vertex w inheriting all other edge incidences involving u and v. Contraction tends to make a graph denser and thus increase edge-bandwidth. In some applications, one restricts attention to simple graphs and thus discards loops or multiple edges that arise under contraction. Such a convention can discard many edges and thus lead to a decrease in edge-bandwidth. In particular, contracting an edge of a clique would yield a smaller clique under this model and thus smaller edge-bandwidth.

For the next result, we say that H is an *elementary contraction* of G if H is obtained from G by contracting one edge and keeping all other edges, regardless of whether loops or multiple edges arise. Edge-bandwidth is a valid parameter for multigraphs.

THEOREM 4.3. *If H is an elementary contraction of G, then $B'(G)-1 \le B'(H) \le 2B'(G) - 1$, and these bounds are sharp for each value of $B'(G)$.*

Proof. Let e be the edge contracted to produce H. For the upper bound, let g be an optimal edge-numbering of G, and let f be the edge-numbering of H produced by deleting e from the numbering. In particular, leave the g-labels below $g(e)$ unchanged and decrement those above $g(e)$ by 1. Edges incident in H have distance at most 2 in $L(G)$, and their distance in $L(G)$ is 2 only if e lies between them. Thus the difference between their g-labels is at most $2B'(g)$, with equality only if the difference between their f-labels is $2B'(G) - 1$.

Equality holds when G is the double-star (the caterpillar with two vertices of degree $k+1$ and $2k$ vertices of degree 1) and e is the central edge of G, so H is the star $K_{1,2k}$. We have observed that $B'(G) = k$ and $B'(H) = 2k - 1$.

For the lower bound, let f be an optimal edge-numbering of H, and let g be the edge-numbering of G produced by inserting e into the numbering just above the edge e' with lowest f-label among those incident to the contracted vertex w in H. In particular, leave f-labels up to $f(e')$ unchanged, augment those above $f(e')$ by 1, and let $g(e) = f(e')+1$. The construction and the argument depend on the preservation of loops and multiple edges. Edges other than e that are incident in G are also incident in H, and the difference between their labels under g is at most one more than the difference under f. Edges incident to e in G are incident to e' in H and thus have f-label at most $f(e') + B'(f)$. Thus their g-label differs from that of e' by at most $B'(f)$.

The lower bound must be sharp for each value of $B'(G)$, because successive contractions eventually eliminate all edges and thus reduce the bandwidth. ∎

5. Edge-bandwidth of cliques and bicliques. We have computed edge-bandwidth for caterpillars and other sparse graphs. In this section we compute edge-bandwidth for classical dense families, the cliques and equipartite complete bipartite graphs. Given the difficulty of bandwidth computations, the existence of exact formulas is of as much interest as the formulas themselves.

THEOREM 5.1. *$B'(K_n) = \lfloor n^2/4 \rfloor + \lceil n/2 \rceil - 2$.*

Proof. Lower bound. Consider an optimal numbering. Among the lowest $\binom{\lceil n/2\rceil - 1}{2} + 1$ values there must be edges involving at least $\lceil n/2 \rceil$ vertices of K_n. Among the highest $\binom{\lfloor n/2 \rfloor}{2} + 1$ values there must be edges involving at least $\lfloor n/2 \rfloor + 1$ vertices of K_n. Since $\lceil n/2 \rceil + \lfloor n/2 \rfloor + 1 > n$, some vertex has incident edges with labels among the lowest $\binom{\lceil n/2\rceil - 1}{2} + 1$ and among the highest $\binom{\lfloor n/2 \rfloor}{2} + 1$. Therefore,

$$\begin{aligned} B'(K_n) &\geq \left[\binom{n}{2} - \binom{\lfloor n/2 \rfloor}{2}\right] - \left[\binom{\lceil n/2 \rceil - 1}{2} + 1\right] \\ &= \left(\left\lceil \frac{n}{2} \right\rceil - 1\right)\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n - 1 - 1 \\ &= \left\lfloor \frac{n^2}{4} \right\rfloor + \left\lceil \frac{n}{2} \right\rceil - 2. \end{aligned}$$

Upper bound. To achieve the bound above, let X, Y be the vertex partition with $X = \{1, \dots, \lceil n/2 \rceil\}$ and $Y = \{\lceil n/2 \rceil + 1, \dots, n\}$. We assign the lowest $\binom{\lceil n/2 \rceil}{2}$ values to the edges within X. We use reverse lexicographic order, listing first the edges with higher vertex 2, then higher vertex 3, etc. We assign the highest $\binom{\lfloor n/2 \rfloor}{2}$ values to the edges within Y by the symmetric procedure.

u	1	1	2	1	2	3	$\cdots$	$\cdots$	$n-3$	$n-3$	$n-3$	$n-2$	$n-2$	$n-1$
v	2	3	3	4	4	4	$\cdots$	$\cdots$	$n-2$	$n-1$	n	$n-1$	n	n
$f(uv)$	1	2	3	4	5	6	$\cdots$	$\cdots$	$\binom{n}{2}-5$					$\binom{n}{2}$

Note that the lowest label on an edge incident to vertex $\lceil n/2 \rceil$ is $1 + \binom{\lceil n/2 \rceil - 1}{2}$.

The labels between these ranges are assigned to the "cross-edges" between X and Y. The cross-edges involving the vertex $\lceil n/2\rceil \in X$ receive the highest of the central labels, and the cross-edges involving $\lceil n/2\rceil+1 \in Y$ (but not $\lceil n/2\rceil$) receive the lowest of these labels. Since the highest cross-edge label is $\binom{n}{2}-\binom{\lfloor n/2\rfloor}{2}$ and the lowest label of an edge incident to $\lceil n/2\rceil$ is $1+\binom{\lceil n/2\rceil-1}{2}$, the maximum difference between labels on edges incident to $\lceil n/2\rceil$ is precisely the lower bound on $B'(K_n)$ computed above. This observation holds symmetrically for the edges incident to $\lceil n/2\rceil+1$.

1 1 2 1 2 3	5 5 5 6 6 7 8 7 8 6 7 8 5 6 7 8	5 5 5 6 6 7
2 3 3 4 4 4	1 2 3 1 2 1 1 2 2 3 3 3 4 4 4 4	6 7 8 7 8 8

We now procede iteratively. On the high end of the remaining gap, we assign the values to the remaining edges incident to $\lceil n/2\rceil-1$. Then on the low end, we assign values to the remaining edges incident to $\lceil n/2\rceil+2$. We continue alternating between the top and the bottom, completing the edges incident to the more extreme labels as we approach the center of the numbering. We have illustrated the resulting order for K_8. Each time we insert the remaining edges incident to a vertex of X, the rightmost extreme moves toward the center at least as much from the previous extreme as the leftmost extreme moves toward the left. Thus the bound on the difference is maintained for the edges incident to each vertex. The observation is symmetric for edges incident to vertices of Y. $\square$

For equipartite complete bipartite graphs, we have a similar construction involving low vertices, high vertices, and cross-edges.

THEOREM 5.2. $B'(K_{n,n})=\binom{n+1}{2}-1$.

Proof. Lower bound. We use the boundary bound of Proposition 2.3 with $k=\lfloor n^2/4\rfloor+1$. Every set of k edges is together incident to at least $n+1$ vertices, since a bipartite graph with n vertices has at most $k-1$ edges. Since $K_{n,n}$ has $2n$ vertices, at most $\lfloor (n-1)^2/4\rfloor$ edges remain when these vertices are deleted. Thus when $|F|=k$, we have

$$B'(K_{n,n}) \geq |\partial(F)| \geq n^2-\left\lfloor\frac{(n-1)^2}{4}\right\rfloor-\left\lfloor\frac{n^2}{4}\right\rfloor-1=\binom{n+1}{2}-1.$$

We construct an ordering achieving this bound. Let $X=\{x_1,\ldots,x_n\}$ and $Y=\{y_1,\ldots,y_n\}$ be the partite sets. Order the vertices as $L=x_1,y_1,\ldots,x_n,y_n$. We alternately finish a vertex from the beginning of L and a vertex from the end. When finishing a vertex from the beginning, we place its incident edges to vertices earlier in L at the end of the initial portion of the numbering f that has already been determined. When finishing a vertex from the end of L, we place its incident edges to vertices later in L at the beginning of the terminal portion of f that has been determined. We do not place an edge twice. When we have finished each vertex in each direction, we have placed all edges in the numbering. For example, this produces the following edge ordering for $K_{6,6}$:

X	1 2 1 2	3 3 1 2 3	4 4 4 1 2 5 1 1 6	2 2 5 6 3 3 3 5 6	4 4 4 5 6	5 5 6 6
Y	1 1 2 2	1 2 3 3 3	1 2 3 4 4 1 5 6 1	5 6 2 2 4 5 6 3 3	4 5 6 4 4	5 6 5 6

It suffices to show that for the jth vertex $v_j \in L$, there are at least $n^2-\binom{n+1}{2}=\binom{n}{2}$ edges that come before the first edge incident to v or after the last edge incident to v. For $j=n+1$, there are exactly $\lfloor n^2/4\rfloor$ edges before the first appearance of v_j and exactly $\lfloor (n-1)^2/4\rfloor$ edges after its last appearance, which matches the argument in the lower bound. As j decreases, the leftmost appearance of v_j moves leftward no more quickly than the rightmost appearance; we omit the numerical details. The symmetric argument applies for $j \geq n$. □

REFERENCES

[1] S. F. Assmann, G. W. Peck, M. M. Sysło, and J. Zak, *The bandwidth of caterpillars with hairs of length* 1 *and* 2, SIAM J. Alg. Discrete Methods, 2 (1981), pp. 387–393.

[2] P. Z. Chinn, J. Chvátalová, A. K. Dewdney, and N. E. Gibbs, *The bandwidth problem for graphs and matrices—a survey*, J. Graph Theory, 6 (1982), pp. 223–254.

[3] F. R. K. Chung, *Labelings of graphs*, in Selected Topics in Graph Theory, III, L. Beineke and R. Wilson, eds., Academic Press, New York, 1988, pp. 151–168.

[4] J. Chvátalová, *Optimal labelling of a product of two paths*, Discrete Math., 11 (1975), pp. 249–253.

[5] J. Chvátalová and J.Opatrný, *The bandwidth problem and operations on graphs*, Discrete Math., 61 (1986), pp. 141–150.

[6] J. Chvátalová and J. Opatrný, *The bandwidth of theta graphs*, Utilitas Math., 33 (1988), pp. 9–22.

[7] D. Eichhorn, D. Mubayi, K. O'Bryant, and D. B. West, *The edge-bandwidth of theta graphs*, to appear.

[8] M. R. Garey, R. L. Graham, D. S. Johnson, and D. E. Knuth, *Complexity results for bandwidth minimization*, SIAM J. Appl. Math., 34 (1978), pp. 477–495.

[9] L. H. Harper, *Optimal assignments of numbers to vertices*, J. Soc. Indust. Appl. Math., 12 (1964), pp. 131–135.

[10] R. Hochberg, C. McDiarmid, and M. Saks, *On the bandwidth of triangulated triangles*, in Proceedings 14th British Combinatorics Conference, Keele, 1993, Discrete Math., 138 (1995), 261–265.

[11] L. T. Q. Hung, M. M. Sysło, M. L. Weaver, and D. B. West, *Bandwidth and density for block graphs*, Discrete Math., 189 (1998), pp. 163–176.

[12] D. J. Kleitman and R. V. Vohra, *Computing the bandwidth of interval graphs*, SIAM J. Discrete Math., 3 (1990), pp. 373–375.

[13] J. H. Mai, *The bandwidth of the graph formed by n meridian lines on a sphere*, J. Math. Res. Exposition, 3 (1983), pp. 55–60 (in Chinese with English summary).

[14] Z. Miller, *The bandwidth of caterpillar graphs*, in Proceedings 12th Southeastern Conference, Congr. Numer., 33 (1981), pp. 235–252.

[15] H. S. Moghadam, *Compression Operators and a Solution to the Bandwidth Problem of the Product of n Paths*, Ph.D. thesis, University of California, Riverside, 1983.

[16] B. Monien, *The bandwidth minimization problem for caterpillars with hair length* 3 *is NP-complete*, SIAM J. Alg. Discrete Methods, 7 (1986), pp. 505–512.

[17] C. H. Papadimitriou, *The NP-completeness of the bandwidth minimization problem*, Computing, 16 (1976), pp. 263–270.

[18] G. W. Peck and A. Shastri, *Bandwidth of theta graphs with short paths*, Discrete Math., 103 (1992), pp. 177–187.

[19] L. Smithline, *Bandwidth of the complete k-ary tree*, Discrete Math., 142 (1995), pp. 203–212.

[20] A. P. Sprague, *An $O(n \log n)$ algorithm for bandwidth of interval graphs*, SIAM J. Discrete Math., 7 (1994), pp. 213–220.

[21] M. M. Sysło and J. Zak, *The bandwidth problem: Critical subgraphs and the solution for caterpillars*, Ann. Discrete Math., 16 (1982), pp. 281–286.

[22] J. F. Wang, D. B. West, and B. Yao, *Maximum bandwidth under edge addition*, J. Graph Theory, 20 (1995), pp. 87–90.

SIAM J. DISCRETE MATH.
Vol. 12, No. 3, pp. 317–325

LINEAR-TIME SUCCINCT ENCODINGS OF PLANAR GRAPHS VIA CANONICAL ORDERINGS*

XIN HE†, MING-YANG KAO‡, AND HSUEH-I LU§

Abstract. Let G be an embedded planar undirected graph that has n vertices, m edges, and f faces but has no self-loop or multiple edge. If G is triangulated, we can encode it using $\frac{4}{3}m - 1$ bits, improving on the best previous bound of about $1.53m$ bits. In case exponential time is acceptable, roughly $1.08m$ bits have been known to suffice. If G is triconnected, we use at most $(2.5 + 2\log 3)\min\{n, f\} - 7$ bits, which is at most $2.835m$ bits and smaller than the best previous bound of $3m$ bits. Both of our schemes take $O(n)$ time for encoding and decoding.

Key words. data compression, graph encoding, canonical ordering, planar graphs, triconnected graphs, triangulations

AMS subject classifications. 05C30, 05C78, 05C85, 68R10

PII. S0895480197325031

1. Introduction.

This paper investigates the problem of *encoding* a given graph G into a binary string S with the requirement that S can be *decoded* to reconstruct G. The problem has been studied generally with two primary objectives. One is to minimize the length of S, while the other is to minimize the time needed to compute and decode S. In light of these goals, a coding scheme is *efficient* if its encoding and decoding procedures both take polynomial time. A coding scheme is *succinct* if the length of S is not much larger than its *information-theoretic tight bound*, i.e., the shortest length over all possible coding schemes.

As the two primary objectives are often in conflict, a number of coding schemes with different trade-offs have been proposed from practical and theoretical perspectives. The most well known efficient succinct scheme is the folklore scheme of encoding a rooted-ordered n-vertex tree into a string of balanced $n-1$ pairs of left and right parentheses, which uses $2(n-1)$ bits. Since the total number of such trees is at least $\frac{1}{2(n-1)} \cdot \frac{(2n-2)!}{(n-1)!(n-1)!}$, the minimum number of bits needed to differentiate these trees is the logarithm[1] of this quantity, which is $2n - o(n)$ by Stirling's approximation. Thus, two bits per edge is an information-theoretic tight bound for encoding rooted-ordered trees. The standard adjacency-list encoding of a graph is widely useful but requires $\Theta(m\log n)$ bits, where m and n are the numbers of edges and vertices, respectively [3]. For certain graph families, Kannan, Naor, and Rudich [10] gave schemes that encode each vertex with $O(\log n)$ bits and support $O(\log n)$-time testing of adjacency between two vertices. For connected planar graphs, Jacobson [9] gave an $\Theta(n)$-bit encoding which supports traversal in $\Theta(\log n)$ time per vertex visited. This result was recently improved by Munro and Raman [17]; their schemes encode binary trees,

*Received by the editors August 8, 1997; accepted for publication (in revised form) February 18, 1999; published electronically September 7, 1999.

http://www.siam.org/journals/sidma/12-3/32503.html.

†Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY 14260 (xinhe@cse.buffalo.edu). The research of this author was supported in part by NSF grant CCR-9205982.

‡Department of Computer Science, Yale University, New Haven, CT 06250-8285 (kao-ming-yang@cs.yale.edu). The research of this author was supported in part by NSF grant CCR-9531028.

§Department of Computer Science and Information Engineering, National Chung-Cheng University, Chia-Yi 621, Taiwan, ROC (hil@cs.ccu.edu.tw).

[1] All logarithms are of base 2.

rooted-ordered trees, and planar graphs succinctly and support several graph operations in constant time. For dense graphs and complement graphs, Kao, Occhiogrosso, and Teng [14] devised two compressed representations from adjacency lists to speed up basic graph techniques such as breadth-first search and depth-first search. Galperin and Wigderson [6] and Papadimitriou and Yannakakis [19] investigated complexity issues arising from encoding a graph by a small circuit that computes its adjacency matrix. For labeled planar graphs, Itai and Rodeh [8] gave an encoding procedure that requires $\frac{3}{2}n \log n + O(n)$ bits. For unlabeled general graphs, Naor [18] gave an encoding of $\frac{n^2}{2} - n \log n + O(n)$ bits, which is optimal to the second order.

Our work aims to minimize the number of bits needed to encode an embedded planar graph G which is unlabeled and undirected. We assume that G has n vertices, m edges, and f faces but has no self-loop nor multiple edge. (See [2, 3, 7, 16] for the graph-theoretic terminology used in this paper.) Note that if polynomial time for encoding and decoding is not required, then any given graph in a large family can be encoded with the information-theoretic minimum number of bits by brute-force enumeration. This paper focuses on schemes that use only $O(n)$ time for both encoding and decoding.

For a general planar graph G, Turán [21] gave an encoding using $4m$ bits asymptotically. This space complexity was improved by Keeler and Westbrook [15] to about $3.58m$ bits. They also gave encoding algorithms for several important classes of planar graphs. In particular, they showed that if G is triangulated, it can be encoded in about $1.53m$ bits. If G is triconnected, it can be encoded using $3m$ bits. In this paper, these latter two results are improved as follows. If G is triangulated, it can be encoded using $\frac{4}{3}m - 1$ bits. It is interesting that rooted-ordered trees require two bits per edge, while the seemingly more complex plane triangulations need fewer bits. Note that Tutte [22] gave an enumeration theorem that yields an information-theoretic tight bound of roughly $1.08m$ bits for plane triangulations that may contain multiple edges. If G is triconnected, we can encode it using at most $(2.5 + 2 \log 3) \min\{n, f\} - 7$ bits, which is at most $2.835m$ bits. Both of our coding schemes are intuitive and simple. They require only $O(n)$ time for encoding as well as decoding. The schemes make new uses of the canonical orderings of planar graphs, which were originally introduced by de Fraysseix, Pach, and Pollack [4] and extended by Kant [11]. These structures and closely related ones have proven useful also for drawing planar graphs in organized and compact manners [12, 13, 20].

This paper is organized as follows. In section 2, we present our coding scheme for plane triangulations. In section 3, we generalize the scheme to encode triconnected plane graphs. We conclude the paper with some open problems in section 4.

2. A coding scheme for plane triangulations. This section assumes that G is a plane triangulation. Thus, $n \geq 3$ and G has $m = 3n - 6$ edges.

Let $v_1, \ldots, v_n$ be an ordering of the vertices of G, where v_1, v_2, v_n are the three exterior vertices of G in the counterclockwise order. After fixing such an ordering, let G_k be the subgraph of G induced by $v_1, \ldots, v_k$. Let H_k be the exterior face of G_k. Let $G - G_k$ be the subgraph of G obtained by removing $v_1, \ldots, v_k$. Our coding scheme uses a special kind of ordering defined as follows.

Definition 2.1 (see [4]). *An ordering $v_1, \ldots, v_n$ of G is* canonical *if the following statements hold for every $k = 3, \ldots, n$:*

1. *G_k is biconnected, and its exterior face H_k is a cycle containing the edge (v_1, v_2).*
2. *The vertex v_k is on the exterior face of G_k, and the set of its neighbors in*

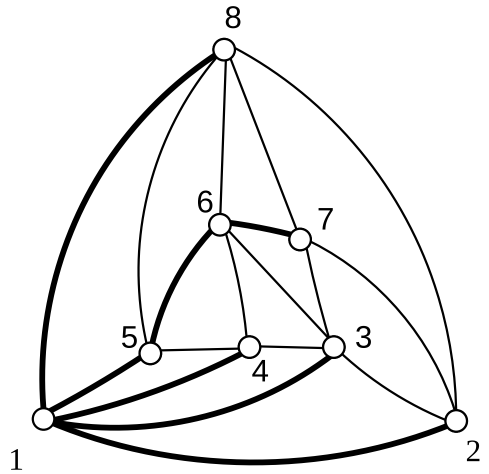


FIG. 2.1. *A plane triangulation and a canonical ordering.*

G_{k-1} forms a subinterval of the path $H_{k-1} - \{(v_1, v_2)\}$ and consists of at least two vertices. Furthermore, if $k < n$, v_k has at least one neighbor in $G - G_k$. Note that the case $k = 3$ is somewhat ambiguous due to degeneracy, and $H_2 - \{(v_1, v_2)\}$ is regarded as the edge (v_1, v_2) itself.

Figure 2.1 illustrates a canonical ordering of a plane triangulation. Note that every plane triangulation has a canonical ordering which can be computed in $O(n)$ time [4]. A canonical ordering of G can be viewed as an order in which G is reconstructed from a single edge (v_1, v_2) step by step. At step k with $3 \leq k \leq n$, the vertex v_k and the edges between v_k and its lower ordered neighbors are added into the graph. For the sake of enhancing intuitions, we call H_{k-1} the *contour* of G_{k-1}; denote its vertices by $c_1(= v_1), c_2, \ldots, c_{t-1}, c_t(= v_2)$ in the consecutive order along the cycle H_{k-1}; and visualize them as arranged from left to right above the edge (v_1, v_2) in the plane. When the vertex v_k is added to G_{k-1} to construct G_k, let $c_\ell, c_{\ell+1}, \ldots, c_r$ be the neighbors of v_k on the contour H_{k-1}. After v_k is added, the vertices $c_{\ell+1}, \ldots, c_{r-1}$ are no longer contour vertices. Thus, we say that these vertices are *covered* by v_k. The edge (v_k, c_ℓ) is the *left* edge of v_k, the edge (v_k, c_r) is the *right* edge of v_k, and the edges (c_p, v_k) with $\ell < p < r$ are the *internal* edges of v_k.

There is no published reference for the following folklore lemma; for the sake of completeness, we include its proof here.

LEMMA 2.2. *Let $v_1, \ldots, v_n$ be a canonical ordering of G. Let T_1 (respectively, T_2) be the collection of the left (respectively, right) edges of v_j for $3 \leq j \leq n-1$; similarly, let T_n be that of the internal edges of v_j for $3 \leq j \leq n$.*

1. *T_1 is a tree spanning over $G - \{v_2, v_n\}$.*
2. *T_2 is a tree spanning over $G - \{v_1, v_n\}$.*
3. *T_n is a tree spanning over $G - \{v_1, v_2\}$.*

Proof. The statements are proved separately as follows.

Statement 1. For $i = 3, \ldots, n-1$, let D_i be the collection of the left edges of v_j for $3 \leq j \leq i$. We prove by induction on i the claim that D_i is a tree spanning over $v_1, v_3, \ldots, v_i$. Then, since $T_1 = D_{n-1}$, the claim implies the statement. For the base case $i = 3$, the claim trivially holds. The induction hypothesis is that the claim holds for $i = k-1 < n-1$. The induction step is to prove the claim for $i = k \leq n-1$.

D_k is obtained from D_{k-1} by adding the left edge (v_k, c_ℓ) of v_k. By the induction hypothesis, D_{k-1} is a tree spanning over $v_1, v_3, \ldots, v_{k-1}$. Since c_ℓ is the left-most neighbor of v_k on H_{k-1}, c_ℓ is some v_j with $1 \le j \le k-1$ and $j \ne 2$. Thus, D_{k-1} contains c_ℓ, and D_k is a tree spanning over $v_1, v_3, \ldots, v_{k-1}, v_k$.

Statement 2. The proof is symmetric to that of Statement 1.

Statement 3. G has n vertices and $3n-6$ edges. The edges $(v_1, v_2), (v_2, v_n), (v_1, v_n)$ are not in $T_1 \cup T_2 \cup T_n$. Thus, since T_1 and T_2 have $n-3$ edges each, T_n has $n-3$ edges. Then, since T_n is acyclic and does not contain v_1 and v_2, T_n is a spanning tree of $G - \{v_1, v_2\}$. □

A canonical ordering $v_1, \ldots, v_n$ is *right-most* if for all v_k and $v_{k'}$ with $k' > k$ such that the neighbors of $v_{k'}$ on $H_{k'-1}$ are all in H_{k-1}, the *left-most* neighbor of $v_{k'}$ appears before that of v_k when traversing H_{k-1} from v_1 to v_2 clockwise. Intuitively speaking, if there are more than one vertex that can be added to G_{k-1}, we always add the right-most one. The ordering in Figure 2.1 is right-most. A right-most canonical ordering is symmetric to a left-most one in [11] and can be computed from G in linear time similarly.

Let $v_1, \ldots, v_n$ be a right-most canonical ordering of G. Let T_1 be as in Lemma 2.2 for this ordering. Let T be the tree $T_1 \cup \{(v_1, v_n), (v_1, v_2)\}$. In Figure 2.1, T is indicated by the thick lines. Our coding scheme uses T extensively. The *right-most* depth-first search of T proceeds as follows. We start at v_1 and traverse the edge (v_1, v_2) first. Afterward, if two or more vertices can be visited from v_k, we choose the right-most one. More precisely, let P be the path in T from v_k to v_1 and then to v_2. Let D be the set of edges between v_k and the available vertices. We visit a new vertex through the edge in D that is next to P in the counterclockwise cyclic order around v_k formed by P and the edges in D. Note that the order in which the vertices are visited by the right-most depth-first search is the right-most canonical ordering $v_1, \ldots, v_n$ that defines T.

We are now ready to describe the encoding S of G as the concatenation of two binary strings S_1 and S_2 as follows.

S_1 is the binary string that encodes T using the folklore parenthesis coding scheme where 0 and 1 correspond to "(" and ")", respectively. In this encoding, T is rooted at v_1, and the branches are ordered the same as their endpoints are in the right-most canonical ordering. Since T contains n vertices, S_1 has $2(n-1)$ bits.

S_2 encodes the number of contour vertices covered by each v_k with $3 \le k \le n$. First, we create a string of $n-2$ copies of 0. The $(k-2)$th 0 corresponds to v_k. If v_k covers d vertices, we insert d copies of 1 before the corresponding 0. For example, the string S_2 for Figure 2.1 is

$$00010101110.$$

Since each vertex v_k with $3 \le k \le n-1$ is covered exactly once, S_2 has $n-3$ copies of 1. So $|S_2| = (n-2) + (n-3) = 2n-5$ bits. Hence, $|S| = |S_1| + |S_2| = 4n-7$ bits.

We next describe how to decode S to reconstruct G. Given S, we can uniquely determine n from the length of S. Subsequently, we can uniquely determine S_1 and S_2. From S_1, we can reconstruct T. From T, we can recover the ordering $v_1, \ldots, v_n$. Then, we draw the edge (v_1, v_2) and perform a loop of $n-2$ steps indexed by k with $3 \le k \le n$, where step k processes v_k. Before v_k is processed, G_{k-1} and its contour H_{k-1} have been constructed. At step k, we add v_k and the edges between v_k and its lower ordered neighbors into G_{k-1} to construct G_k as follows. From T, we can identify the left-most neighbor c_ℓ of v_k on the contour H_{k-1} because c_ℓ is simply the parent of v_k in T. From S_2, we can determine the number d of vertices covered by v_k. Thus, we add the edges $(c_\ell, v_k), (c_{\ell+1}, v_k), \ldots, (c_{\ell+d+1}, v_k)$ into G_{k-1}; note that $r = \ell + d + 1$. This gives us the subgraph G_k and completes step k.

It is straightforward to carry out these encoding and decoding procedures in linear time. Also, we can save one bit by deleting the last 0 in S_2. Since v_3 covers no vertex, for $n \geq 4$, we can save another bit by deleting the first 0 in S_2. Note that for $n = 3$, the last 0 in S_2 is also the first 0 and cannot be deleted twice, but we can simply encode the 3-vertex plane triangulation with zero bit without ambiguity. Thus, we have the following theorem.

THEOREM 2.3. *A plane triangulation of m edges and n vertices with $n \geq 4$ can be encoded using $4n - 9 = \frac{4}{3}m - 1$ bits. Both encoding and decoding take $O(n)$ time.*

3. A coding scheme for triconnected plane graphs. This section assumes that G is triconnected. To avoid triviality, let $n \geq 3$.

Let $v_1, \ldots, v_n$ be an ordering of the vertices of G, where v_1, v_2, v_n are on the exterior face of G, and v_2 and v_n are neighbors of v_1. Let G_k be the subgraph of G induced by $v_1, \ldots, v_k$. Let H_k be the exterior face of G_k. Let $G - G_k$ be the subgraph of G obtained by removing $v_1, \ldots, v_k$. Our coding scheme for triconnected plane graphs uses an ordering defined as follows.

DEFINITION 3.1 (see [11]). *An ordering $v_1, \ldots, v_n$ of a triconnected plane graph G is* canonical *if the integer interval $[3, n]$ can be partitioned into subintervals $[k, k+q]$ each satisfying either set of properties below:*

1. *The integer q is 0. The vertex v_k is on the exterior face of G_k and has at least two neighbors in G_{k-1}. G_k is biconnected and its exterior face contains the edge (v_1, v_2). If $k < n$, v_k has at least one neighbor in $G - G_k$.*
2. *The integer q is at least 1. The sequence $v_k, v_{k+1}, \ldots, v_{k+q}$ is a chain on the exterior face of G_{k+q} and has exactly two neighbors in G_{k-1}, one for v_k and the other for v_{k+q}, which are on the exterior face of G_{k-1}. G_{k+q} is biconnected and its exterior face contains the edge (v_1, v_2). Every vertex among $v_k, \ldots, v_{k+q}$ has at least one neighbor in $G - G_{k+q}$.*

As in section 2, we similarly define a right-most canonical ordering $v_1, \ldots, v_n$ of G. Figure 3.1 shows a right-most canonical ordering of a triconnected plane graph. Given a triconnected plane graph, we can find a right-most canonical ordering in linear time [11]. With a right-most canonical ordering, G can be reconstructed from a single edge (v_1, v_2) through a sequence of steps indexed by k'. There are two possible cases at step k', which correspond to the two sets of properties in Definition 3.1 and are used throughout this section.

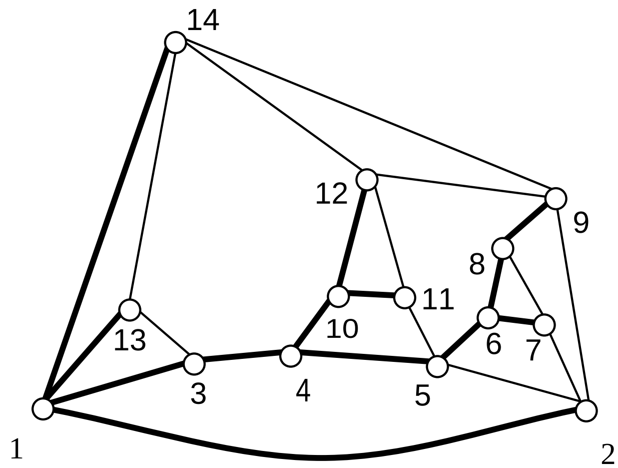
Case 1. A single vertex v_k is added.

Case 2. A chain of $q + 1$ vertices $v_k, \ldots, v_{k+q}$ is added.

While reconstructing G, we collect a set T of edges as follows. Initially, T consists of the edge (v_1, v_2). Let $c_1(= v_1), c_2, \ldots, c_{t-1}, c_t(= v_2)$ be the vertices of H_{k-1}, which are ordered consecutively along the boundary cycle of H_{k-1} and are arranged from left to right above the edge (v_1, v_2) in the plane.

Case 1. Let c_ℓ and c_r with $1 \leq \ell < r \leq t$ be the left-most and right-most neighbors of v_k in H_{k-1}, respectively. After v_k is added, $c_{\ell+1}, \ldots, c_{r-1}$ are no longer contour vertices; these vertices are *covered* at step k'. The edge (c_ℓ, v_k) is included in T.

Case 2. Let c_ℓ and c_r with $1 \leq \ell < r \leq t$ be the neighbors of v_k and v_{k+q} in H_{k-1}, respectively. After $v_k, \ldots, v_{k+q}$ are added, $c_{\ell+1}, \ldots, c_{r-1}$ are no longer contour vertices; these vertices are covered at step k'. The edges $(c_\ell, v_k), (v_k, v_{k+1}), \ldots, (v_{k+q-1},$



Step:	Vertices added:
1	3, 4, 5
2	6, 7
3	8
4	9
5	10, 11
6	12
7	13
8	14

FIG. 3.1. *A triconnected plane graph and a canonical ordering.*

v_{k+q}) are included in T.

In Figure 3.1, the edges in T are indicated by the thick lines. By an argument similar to the proof of Lemma 2.2, Statement 1, T is a spanning tree of G. As in section 2, we similarly define the *right-most* depth-first search in T. Note that the order in which the vertices of T are visited by the right-most depth-first search is the right-most canonical ordering $v_1, \ldots, v_n$ that defines T.

We are now ready to describe the encoding S of G by means of T. We further divide Case 1 into three subcases.

Case 1a. No vertex is covered at step k'.

Case 1b. At least one vertex is covered at step k' and the left-most covered vertex $c_{\ell+1}$ is adjacent to v_k.

Case 1c. At least one vertex is covered at step k' and the left-most covered vertex $c_{\ell+1}$ is not adjacent to v_k.

Let β be the number of steps for reconstructing G. Let $\beta_{1a}, \beta_{1b}, \beta_{1c}$, and β_2 be the numbers of steps of Cases 1a, 1b, 1c, and 2, respectively. We first consider the case $\beta_{1b} \geq \beta_{1c}$ to encode G with *Scheme* I; afterwards, we modify Scheme I into *Scheme* II for the case $\beta_{1b} < \beta_{1c}$.

In Scheme I, the encoding S of G is the concatenation of three strings S_1, S_2, and S_3. S_1 is the folklore parentheses encoding of T, which is rooted and ordered in the same way as in section 2. Since T has n vertices, S_1 has $2(n-1)$ bits.

To construct S_2, first let $Q = s_1 * s_2 * \cdots * s_\beta *$, where each $s_{k'}$ is a binary string that corresponds to the step k' of reconstructing G based on the ordering $v_1, \ldots, v_n$. $s_{k'}$ is determined as follows. The following two cases both assume that d vertices are covered at step k'.

Case 1. Note that $d = r - \ell - 1$. The string $s_{k'}$ has d symbols corresponding to c_j with $j = \ell+1, \ldots, r-1$, respectively. If the edge (c_j, v_k) is present in G, the symbol in $s_{k'}$ corresponding to c_j is 1; otherwise, the symbol is 0. Note that in Case 1a, since no vertex is covered, $s_{k'}$ is empty.

Case 2. The string $s_{k'}$ consists of q copies of 0 followed by d copies of 1. For example, the string Q for Figure 3.1 is

$$\begin{array}{cccccccccccccc} \underbrace{00} & * & 0 & * & * & 0 & * & 0 & * & \underbrace{1000} & * & * & \underbrace{10001} & * \\ \uparrow & & \uparrow & & & \uparrow & & \uparrow & & \uparrow & & & \uparrow & \\ s_1 & & s_2 & & & s_4 & & s_5 & & s_6 & & & s_8 & \end{array}$$

S_2 is a binary representation of Q defined as follows. A step of Case 1 adds one vertex to G and correspondingly includes one $*$ in Q; similarly, a step of Case 2 adds $q+1$ vertices to G and includes one $*$ and q copies of 0 in Q. Since exactly $n-2$ vertices are added, the total number of these symbols is $n-2$. Each symbol in Q not yet counted corresponds to a vertex covered at the β steps. Since each v_k with $3 \leq k \leq n-1$ is covered at most once and v_1, v_2, v_n are never covered, the total number of these latter symbols is at most $n-3$. Thus Q has at most $2n-5$ symbols. For the sake of unambiguous decoding, we pad Q with copies of 1 at its end to have exactly $2n-5$ symbols. Since Q uses three distinct symbols, we treat it as an integer of base 3 and convert it to a binary integer. Again, for the sake of unambiguous decoding, we use exactly $\lceil (2n-5)\log 3 \rceil$ bits for this binary integer by padding copies of 0 at its beginning. The resulting binary string is the desired S_2.

For the sake of decoding, we also need to know whether any given $s_{k'}$ is of Cases 1 or 2. Thus, let $S_3 = t_1 \cdots t_\beta$, where $t_{k'} = 1$ if step k' is of Case 1 and $t_{k'} = 0$ otherwise. To save space, note that some bits $t_{k'}$ can be deleted as follows without incurring ambiguity. If step k' is of Case 1a, $t_{k'}$ is deleted because $s_{k'}$ is empty and only a string of Case 1a can be empty. If step k' is of Case 1b, $t_{k'}$ is deleted because $s_{k'}$ starts with 1, while the strings of Case 2 start with 0. If step k' is of Case 1c or 2, $t_{k'}$ remains in S_3. For example, the string S_3 for Figure 3.1 consists of $t_1 = 0$, $t_2 = 0$, $t_4 = 1$, $t_5 = 0$. Thus, S_3 has $\beta_{1c} + \beta_2$ bits, which can be bounded as follows. A step of Case 1 adds one vertex into G and a step of Case 2 adds at least two vertices. Since $n-2$ vertices are added over the β steps, $\beta_{1a} + \beta_{1b} + \beta_{1c} + 2\beta_2 \leq n-2$. Since Scheme I assumes $\beta_{1b} \geq \beta_{1c}$, $|S_3| = \beta_{1c} + \beta_2 \leq \frac{1}{2}\cdot(\beta_{1b} + \beta_{1c}) + \beta_2 \leq \frac{1}{2}\cdot(\beta_{1a} + \beta_{1b} + \beta_{1c} + 2\beta_2) \leq 0.5n-1$.

Since $S = S_1//S_2//S_3$, $|S| \leq 2(n-1) + \lceil (2n-5)\log 3 \rceil + 0.5n - 1 \leq (2.5 + 2\log 3)n - 9$ bits. This completes the description of the encoding procedure of Scheme I.

Next we describe how to decode S to reconstruct G. This decoding assumes that both S and n are given. Thus, we can uniquely determine S_1, S_2, and S_3. Then we convert S_2 to Q. From Q we can recover all $s_{k'}$ with $1 \leq k' \leq \beta$. From S_3 and all $s_{k'}$, we can recover all $t_{k'}$ with $1 \leq k' \leq \beta$. From S_1, we reconstruct T. From T, we find the ordering $v_1, \ldots, v_n$. Afterwards, we draw the edge (v_1, v_2) and perform a loop of steps as follows. Each step is indexed by k' and corresponds to step k' of reconstructing G using the right-most canonical ordering.

If $t_{k'} = 1$, step k' is of Case 1. Thus, a vertex v_k is added at this step, where v_k is the smallest ordered vertex not added into the current graph yet. From T, we can determine the left-most neighbor c_ℓ of v_k in the contour H_{k-1} because c_ℓ is the parent of v_k in T. From $s_{k'}$, we know the number of vertices covered by v_k and hence the right-most neighbor c_r of v_k in the contour H_{k-1}. From $s_{k'}$, we also know which of the covered vertices are connected to v_k. These corresponding edges are added to G.

If $t_{k'} = 0$, step k' is of Case 2. Thus, a chain $v_k, \ldots, v_{k+q}$ is added at this step, where v_k is the smallest ordered vertex not added into the current graph yet. The integer q can be determined from the string $s_{k'}$ by counting its leading copies of 0. From $s_{k'}$, we also know the number of vertices covered at step k', which is the count of 1 in $s_{k'}$. Thus, we know the neighbor c_r of v_{k+q} in the contour H_{k-1}. The chain is added accordingly.

This completes the decoding procedure of Scheme I. It is straightforward to implement the whole Scheme I in $O(n)$ time. If $\beta_{1b} < \beta_{1c}$, we use Scheme II to encode G, which is identical to Scheme I with the following differences. If step k' is of Case 2, $s_{k'}$ consists of q copies of 1 followed by d copies of 0. Also, all bits $t_{k'}$ for steps of Cases 1a and 1c are omitted from S_3 without incurring ambiguity since their corresponding strings $s_{k'}$ are either empty or start with 0, while the strings of Cases 1b and 2 start with 1. We use one extra bit to encode whether we use Scheme I or II. Thus we have the following lemma.

LEMMA 3.2. *Any triconnected plane graph with n vertices can be encoded using at most $(2.5 + 2\log 3)n - 8$ bits. Both encoding and decoding take $O(n)$ time. The decoding procedure assumes that both S and n are given.*

We can improve Lemma 3.2 as follows. Let G^* be the dual of G. G^* has f vertices, m edges and n faces. Since G is triconnected, G^* is also triconnected. Furthermore, if $n > 3$, then $f > 3$ and G^* has no self-loop nor multiple edge. Thus, we can use the coding scheme of Lemma 3.2 to encode G^* with at most $(2.5+2\log 3)f-8$ bits. Since G can be uniquely determined from G^*, to encode G, it suffices to encode G^*. To make S shorter, for the case $n > 3$, if $n \leq f$, we encode G using at most $(2.5 + 2\log 3)n - 8$ bits; otherwise, we encode G^* using at most $(2.5+2\log 3)f-8$ bits. This new encoding has at most $(2.5 + 2\log 3)\min\{n, f\} - 8$ bits. Since $\min\{n, f\} \leq \frac{n+f}{2}$, the bit count is at most $(1.25 + \log 3)m - 2$ by Euler's formula $n + f = m + 2$. For the sake of decoding, we use one extra bit to denote whether we encode G or its dual. Note that if $n = 3$, we can simply encode G using zero bit without ambiguity. Thus we have proved the following theorem.

THEOREM 3.3. *Any triconnected plane graph with n vertices, m edges and f faces can be encoded using at most $(2.5 + 2\log 3)\min\{n, f\} - 7 \leq (1.25 + \log 3)m - 1$ bits. Both encoding and decoding take $O(n)$ time. The decoding procedure assumes that S is given together with n or f as appropriate.*

Remark. There are several ways to improve this coding scheme so that the decoding does not require n as input. One is to use well-known data compression techniques to encode n and append it to the beginning of S using $\log n + O(\log\log n)$ bits [1, 5]. Another is to pad S with copies of 1 at its end so that it has exactly $\lceil(2.5+2\log 3)\min\{n, f\}\rceil - 7$ bits. Then, since $2.5+2\log 3 > 1$, given S alone, we can uniquely determine n or f and proceed with the original decoding procedure. With the strings $s_{k'}$, we can unambiguously identify the padded bits.

4. Open problems. This paper leaves several problems open. Since plane triangulations are useful in many application areas, it would be particularly helpful to encode them in $O(n)$ time using close to $1.08m$ bits. Similarly, it would be significant to obtain a linear-time coding scheme for triconnected plane graphs using close to $2m$ bits. Note that Tutte [23] proved an information-theoretic tight bound of $2m + o(m)$ bits for triconnected plane graphs that may contain multiple edges and self-loops. More generally, it would be of interest to encode graphs in a given family in polynomial time using their information-theoretic minimum number of bits. Solving these problems will most likely lead to the discovery of new structural properties of graphs.

Acknowledgment. The authors are grateful to anonymous referees for helpful comments.

REFERENCES

[1] T. C. BELL, J. G. CLEARY, AND I. H. WITTEN, *Text Compression*, Prentice-Hall, Englewood Cliffs, NJ, 1990.

[2] C. Berge, *Graphs*, 2nd ed., North-Holland, New York, NY, 1985.
[3] T. H. Cormen, C. L. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1990.
[4] H. de Fraysseix, J. Pach, and R. Pollack, *How to draw a planar graph on a grid*, Combinatorica, 10 (1990), pp. 41–51.
[5] P. Elias, *Universal codeword sets and representations of the integers*, IEEE Trans. Inform. Theory, IT-21 (1975), pp. 194–203.
[6] H. Galperin and A. Wigderson, *Succinct representations of graphs*, Inform. and Control, 56 (1983), pp. 183–198.
[7] F. Harary, *Graph Theory*, Addison-Wesley, Reading, MA, 1972.
[8] A. Itai and M. Rodeh, *Representation of graphs*, Acta Inform., 17 (1982), pp. 215–219.
[9] G. Jacobson, *Space-efficient static trees and graphs*, in Proceedings of the 13th Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE Press, Washington, DC, 1989, pp. 549–554.
[10] S. Kannan, M. Naor, and S. Rudich, *Implicit representation of graphs*, SIAM J. Discrete Math., 5 (1992), pp. 596–603.
[11] G. Kant, *Drawing planar graphs using the lmc-ordering*, in Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE Press, Washington, DC, 1992, pp. 101–110.
[12] G. Kant and X. He, *Regular edge labeling of 4-connected plane graphs and its applications in graph drawing problems*, Theoret. Comput. Sci., 172 (1997), pp. 175–193.
[13] M. Y. Kao, M. Fürer, X. He, and B. Raghavachari, *Optimal parallel algorithms for straight-line grid embeddings of planar graphs*, SIAM J. Discrete Math., 7 (1994), pp. 632–646.
[14] M. Y. Kao, N. Occhiogrosso, and S. H. Teng, *Simple and efficient compression schemes for dense and complement graphs*, J. Combin. Optim., (1999), to appear.
[15] K. Keeler and J. Westbrook, *Short encodings of planar graphs and maps*, Discrete Appl. Math., 58 (1995), pp. 239–252.
[16] L. Lovász, *An Algorithmic Theory of Numbers, Graphs and Convexity*, CBMS-NSF Regional Conference Series in Appl. Math. 50, SIAM, Philadelphia, PA, 1986.
[17] J. I. Munro and V. Raman, *Succinct representation of balanced parentheses, static trees and planar graphs*, in Proceedings of the 38th Annual IEEE Symposium on the Foundations of Computer Science (FOCS), IEEE Press, Washington, DC, 1997, pp. 118–126.
[18] M. Naor, *Succinct representations of general unlabeled graphs*, Discrete Appl. Math., 28 (1990), pp. 303–307.
[19] C. H. Papadimitriou and M. Yannakakis, *A note on succinct representations of graphs*, Inform. and Control, 71 (1986), pp. 181–185.
[20] W. Schnyder, *Embedding planar graphs on the grid*, in Proceedings of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, PA, 1990, pp. 138–148.
[21] G. Turán, *On the succinct representation of graphs*, Discrete Appl. Math., 8 (1984), pp. 289–294.
[22] W. T. Tutte, *A census of planar triangulations*, Canad. J. Math., 14 (1962), pp. 21–38.
[23] W. T. Tutte, *A census of planar maps*, Canad. J. Math., 15 (1963), pp. 249–271.

SIAM J. DISCRETE MATH.
Vol. 12, No. 3, pp. 326–336

NEW FACETS OF THE LINEAR ORDERING POLYTOPE*

G. BOLOTASHVILI†, M. KOVALEV†, AND E. GIRLICH‡

Abstract. The linear ordering problem has many applications and was studied by many authors (for a survey, see [P. Fishburn, *SIAM J. Discrete Math.*, 4 (1990), pp. 478–488; M. Grötschel, M. Jünger, and G. Reinelt, *Math. Programming*, 33 (1985), pp. 43–60; G. Reinelt, *The Linear Ordering Problem: Algorithm and Applications*, Heldermann Verlag, 1985]). One approach to solving this problem, the so-called cutting plane method [M. Grötschel, M. Jünger, and G. Reinelt, *Oper. Res.*, 32 (1984), pp. 1195–1220; V. A. Yemelichev, M. M. Kovalev, and M. K. Kravtsov, *Polytopes, Graphs, Optimization,* Cambridge University Press, 1984], derives facet-defining inequalities which are violated by current nonfeasible solution and adds them to the system of inequalities of current linear programming problems.

We present a method (rotation method) for generating new facets of polyhedra by using previously known ones. The rotation method for the linear ordering polytope generalizes facets induced by subgraphs called m-fences, Möbius ladders, and Z_m-facets introduced by Reinelt, (m, k)-fences introduced by Bolotashvili [G. C. Bolotashvili, *A Class of Facets of the Permutation Polytope and a Method for Constructing Facets of the Permutation Polytope*, preprint VINITI N3403-B87, Moscow, 1987 (in Russian)]; and t-reinforced m-fences introduced by Leung and Lee [J. Leung and J. Lee, *Discrete Appl. Math.*, 50 (1984), pp. 185–200]. We introduce 10 collections of inequalities representing facets of the linear ordering polytope. Among them are three that coincide with earlier known ones: m-wheel-facets introduced by Reinelt, augmented m-fences introduced by McLennan [A. McLennan, in *Preferences, Uncertainty and Optimality*, West View Press, 1990, pp. 187–202]; and augmented t-reinforced m-fences introduced by Leung and Lee.

Key words. linear ordering polytope, facets, linear ordering, ranking

AMS subject classifications. Primary, 52B12; Secondary, 90B10

PII. S0895480196300145

1. Introduction. A linear ordering of an n-element set N is a bijection $\pi : \{1, 2, \ldots, n\} \to N$. The *linear ordering polytope* P_n is a convex hull of $n!$ points in R^{n^2-n}. Each of these points corresponds one to one to some $\pi = (\pi(1), \ldots, \pi(n))$ by the following rule: $x_{ij}^{\pi} = 1$ if $\pi^{-1}(i) < \pi^{-1}(j)$ and $x_{ij}^{\pi} = 0$ if $\pi^{-1}(i) > \pi^{-1}(j), i \neq j$.

Let $G = (N, A)$ be a complete directed graph (digraph) with node set N and arc set $A = N \times N$ (without loops). A directed subgraph (N, T) is a *spanning tournament* if for every pair of distinct nodes $u, v \in N$ exactly one of the arcs (u, v) and (v, u) is in T. Given a linear ordering π of the nodes N of a digraph, the arc set $\{(u, v) : \pi^{-1}(u) < \pi^{-1}(v)\}$ forms an *acyclic spanning tournament* on N; conversely, an acyclic spanning tournament (N, T) induces a unique ordering of N. Thus the linear ordering polytope P_n is the convex hull of the incidence vectors of the acyclic spanning tournaments on N.

The system of inequalities and equations

$$x_{ij} \leq 1, \qquad i \neq j, \quad i, j \in N, \tag{1.1}$$

*Received by the editors March 1, 1996; accepted for publication (in revised form) March 19, 1999; published electronically September 7, 1999. This research was partially supported by the Fundamental Research Foundation of Republic Belarus and DAAD.

http://www.siam.org/journals/sidma/12-3/30014.html

†Faculty of Applied Mathematics and Informatics, State University of Belarus, 220080, Minsk, Belarus (kovalev@fpm.bsu.unibel.by).

‡Faculty of Mathematics, Otto-von-Guericke-University, PSF 4120, 39106 Magdeburg, Germany (eberhard.girlich@mathematik.uni-magdeburg.de).

$$x_{ij} + x_{jk} + x_{ki} \leq 2, \qquad i \neq j, \quad i \neq k, \quad j \neq k, \quad i,j,k \in N, \tag{1.2}$$
$$x_{ij} + x_{ji} = 1, \qquad i \neq j, \quad i,j \in N, \tag{1.3}$$

defines the *relaxation polytope* B_n of the polytope P_n. Every point x^π is a vertex of B_n. In addition to these integer vertices, the polytope B_n for $n \geq 6$ has noninteger vertices (note that in earlier publications it was believed that $P_n = B_n$ (cf. [4])). The simplest classes of noninteger vertices are [1]

$$\begin{aligned}(1) \qquad & x^0_{i_s i_q} = x^0_{j_s j_q} = \frac{1}{2},\\ & x^0_{i_s j_q} = 0, x^0_{j_q i_s} = 1, \quad s \neq q, 1 \leq s,q \leq m,\\ & x^0_{i_s j_s} = x^0_{j_s i_s} = \frac{1}{2}, \quad 1 \leq s \leq m,\end{aligned}$$

and

$$\begin{aligned}(2) \qquad & x^{00}_{i_s i_q} = x^{00}_{j_s j_q} = \frac{1}{2}, \quad s \neq q, 1 \leq s,q \leq m,\\ & x^{00}_{i_s j_q} = 0, x^{00}_{j_q i_s} = 1, \quad (j_q, i_s) \in B,\\ & x^{00}_{i_s j_q} = x^{00}_{j_q i_s} = \frac{1}{2}, \quad (j_q, i_s) \notin B,\end{aligned}$$

where $I = (i_1, \ldots, i_m), J = (j_1, \ldots, j_m)$ are arbitrary disjoint subsets of N, $3 \leq m \leq \frac{n}{2}$, k is such that $m + 1 \equiv 0 \pmod{k}$, and

$$B = \bigcup_{s=1}^{m} \bigcup_{q=1}^{k-1} \{(j_{s+q}, i_s), (j_{s-q}, i_s)\}, \quad j_{m+p} = j_p, j_{1-p} = j_{m-p+1}, p = 1, \ldots, m.$$

It is proved in [1, 2, 12] that the inequality

$$\sum_{s=1}^{m} x_{i_s j_s} - \sum_{s=1}^{m} \sum_{\substack{q=1\\ q \neq s}}^{m} x_{i_s j_q} \leq 1 \tag{1.4}$$

cuts off the noninteger vertex x^0 and defines a facet of P_n, whereas the inequality

$$\sum_{s=1}^{m} x_{i_s j_s} - \sum_{s=1}^{m} \sum_{q=1}^{k-1} (x_{i_s j_{s+q}} + x_{i_s j_{s-q}}) \leq \frac{m+1}{k} - 1 \tag{1.5}$$

cuts off the noninteger vertex x^{00} and defines a facet of P_n [2].

Every facet-defining inequality for P_n can be represented in normal form [12, 13], i.e., in a form with nonnegative coefficients. Indeed (since $x_{ij}+x_{ji} = 1$), facet-defining inequality (1.4) can be rewritten as

$$x(F) = \sum_{s=1}^{m} x_{i_s j_s} + \sum_{s=1}^{m} \sum_{\substack{q=1\\ q \neq s}}^{m} x_{j_q i_s} \leq m^2 - m + 1, \tag{1.6}$$

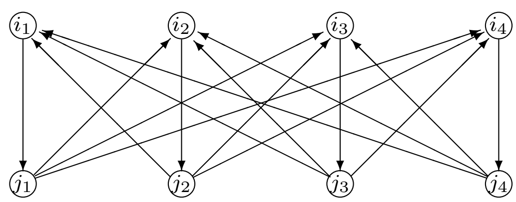
where $F = \bigcup_{s=1}^{m} (\{(i_s, j_s)\} \bigcup \{\bigcup_{q=1}^{m} \{(j_q, i_s) : q \neq s\}\})$.

In this form it has been constructed in [12]. Inequality (1.5) with nonnegative coefficients has the following form:

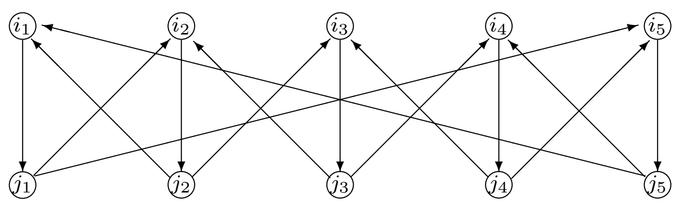
$$x(F') = \sum_{s=1}^{m} x_{i_s j_s} + \sum_{s=1}^{m}\sum_{q=1}^{k-1}(x_{j_{s+q} i_s} + x_{j_{s-q} i_s}) \leq \frac{m+1}{k} - 1 + 2m(k-1), \tag{1.7}$$

where $F' = B \bigcup \{\bigcup_{s=1}^{m}(i_s, j_s)\}$.

If an inequality $ax \leq a_0$ is in normal form, then it induces a weighted digraph corresponding to the nonzero coefficients of the inequality and with $w(i,j) = a_{ij}$, where $w(i,j)$ is the weight of arc (i,j) in the digraph. Conversely, a weighted digraph can be understood to induce an inequality by associating a coefficient of $w(i,j)$ with every arc in the digraph, 0 for all other arcs, and an appropriately defined right-hand side. Digraphs induced by facet-defining inequalities (1.6) and (1.7) are called m-fences and (m,k)-fences, respectively.



a) 4-fence



b) (5,2)-fence

FIG. 1.

Figure 1(a) illustrates an m-fence for $m = 4$; Figure 1(b) illustrates an (m,k)-fence for $m = 5,\ k = 2$. Evidently, if m is odd, then an m-fence is a special case of an (m,k)-fence when $m + 1 = 2k$. Throughout the figures in the paper, arcs shown without numerical labels should be interpreted as having weight equal to 1.

Using substitution $x_{ij} = 1 - x_{ji}$, which follows from (1.3), we can replace every facet-defining inequality $ax \leq b$ with the equivalent inequality $\bar{b} \leq \bar{a}x$. For example, inequality (1.4) can be rewritten as

$$m - 1 \leq \sum_{s=1}^{m} x_{j_s i_s} + \sum_{\substack{s=1 \\ s \neq q}}^{m} \sum_{q=1}^{m} x_{i_q j_s}, \tag{1.8}$$

and inequality (1.5) can be rewritten as

$$m - \frac{m+1}{k} + 1 \leq \sum_{s=1}^{m} x_{j_s i_s} + \sum_{s=1}^{m}\sum_{q=1}^{k-1}(x_{i_s j_{s+q}} + x_{i_s j_{s-q}}). \tag{1.9}$$

LEMMA 1.1 (trivial lifting lemma [12, 13]). *Facet-defining inequalities of P_n also define facets for P_{n_1}, $n_1 > n$.*

2. The rotation method. We present now a new *rotation method* for generating new facets of the linear ordering polytope. The idea for this method was introduced in [3].

Let P be a polyhedron in R^n with the set of vertices $vert\ P$ and ψ be an affine mapping of R^n into itself. If $vert\ P = vert\ \psi(P)$, then ψ is called a *rotation mapping* of P. Evidently, the rotation mappings realize one-to-one mappings of the facet set and corresponding facet-defining inequalities onto themselves. Hence, having a facet-defining inequality $ax \leq a_0$ for the polyhedron P, we have a facet-defining inequality $a\psi(x) \leq a_0$ for the same polyhedron P.

Remark 2.1. If a facet F is defined as $conv\{x^1, \ldots, x^s\}$, then $\psi(F) = conv\{\psi(x^1), \ldots, \psi(x^s)\}$.

A trivial rotation mapping for linear ordering polytope P_n is an *arc reversal* mapping which is defined as $\bar{x}_{ij} = x_{ji}$. This mapping transforms every facet $ax \leq a_0$ into the facet $bx \leq a_0$, where $b_{ij} = a_{ji}$ for all $1 \leq i, j \leq n$ [12, 13]. The trivial rotation mapping does not generate new facets for the most known facets of the polytope P_n, because in many cases the mapping generates another member of the same facet family, i.e., the corresponding digraphs are isomorphic (for example, an m-fence maps to an m-fence, and so on).

For a given vertex $r \in N$, we introduce a mapping $\psi_r : R^{n^2-n} \to R^{n^2-n}$, defined as

$$\bar{x}_{rj} = x_{jr},\ \bar{x}_{jr} = x_{rj},\ j \neq r,\ j \in N, \tag{2.1}$$

$$\bar{x}_{ij} = x_{ij} + x_{jr} - x_{ir},\ i \neq j,\ i \neq r,\ j \neq r,\ i, j \in N. \tag{2.2}$$

An equivalent version of (2.2) using (1.3) is

$$\bar{x}_{ij} = x_{ij} + x_{jr} + x_{ri} - 1,\ i \neq j,\ i \neq r,\ j \neq r, \quad i, j \in N.$$

Observe that this mapping is not a rotation in the strict sense of the word; thus the term rotation mapping is not restricted to mappings that are rotational in the conventional sense but can include reflections as well.

Remark 2.2. The rotation ψ_r of the linear ordering $1, 2, \ldots, r-1, r, r+1, \ldots, n$ maps to the ordering $r+1, \ldots, n, r, 1, \ldots, r-1$.

THEOREM 2.1. *The mapping ψ_r is a rotation mapping of the linear ordering polytope for every $r \in N$.*

Proof. To prove that ψ_r is a rotation mapping for P_n it is sufficient to show that ψ_r transforms the relaxation polytope B_n into itself and all its integer vertices into integer vertices. Indeed, since ψ_r is a nonsingular affine mapping, there exists the inverse mapping ψ_r^{-1}. The mapping ψ_r^{-1} is defined by the following equalities:

$$x_{rj} = \bar{x}_{jr},\ x_{jr} = \bar{x}_{rj},\ x_{ij} = \bar{x}_{ij} + \bar{x}_{jr} - \bar{x}_{ir}, \qquad i \neq j, i \neq r, j \neq r, i, j, r \in N. \tag{2.3}$$

Therefore, the polytope $\psi_r(B_n)$ is defined by conditions

$$\begin{aligned} &1 \geq x_{ij} = \bar{x}_{ij} + \bar{x}_{jr} - \bar{x}_{ir}, \qquad i \neq j, i \neq r, j \neq r;\ i, j, r \in N,\\ &2 \geq x_{ij} + x_{jr} + x_{ri} = \bar{x}_{ij} + \bar{x}_{jr} - \bar{x}_{ir} + \bar{x}_{rj} + \bar{x}_{ir} = \bar{x}_{ij} + 1,\\ &1 = x_{ij} + x_{ji} = \bar{x}_{ij} + \bar{x}_{ji}, \end{aligned}$$

which combined with the equalities $x_{ij} + x_{ji} = 1$ in coordinates $\bar{x}_{ij}$ are identical to (1.1)–(1.3), that is, $\psi_r(B_n) = B_n$. Equalities (2.1)–(2.2) and (2.3) imply that

x is an integer point of B_n if and only if $\bar{x}$ is an integer point of $\psi_r(B_n)$, that is, $vert(\psi_r(B_n)) \cap Z^{n(n-1)} = vert(B_n) \cap Z^{n(n-1)}$, or $vert(\psi_r(P_n)) = vert(P_n)$. □

Theorem 2.1 and the definition of rotation mapping ψ_r directly imply the following technique for obtaining new facets of the linear ordering polytope from known ones.

THEOREM 2.2. *If the inequality* $\sum\sum a_{ij}x_{ij} \leq a_0$ *defines a facet for* P_n, *then the inequality*

$$\sum_{\substack{i=1\\ i\neq r}}^{n} \left(\sum_{\substack{j=1\\ j\neq r}}^{n} a_{ij}(x_{ij} + x_{jr} - x_{ir}) + a_{ir}x_{ri} + a_{ri}x_{ir} \right) \leq a_0$$

defines a facet for P_n.

We distinguish two cases of rotation mapping ψ_r: the first, when the variables x_{ir} and x_{rj} are involved in the facet-defining inequality $\sum\sum a_{ij}x_{ij} \leq a_0$ and the second when they are not. In the first case we'll speak about facet routing, and in the second case, about facet lifting.

Let a facet-defining inequality $ax \leq b$ be in normal form and $G = (N(U), U)$ be a subdigraph of a complete digraph $G_n = (N, A)$ and arc set $U = \{(i, j) \in A \;:\; a_{ij} > 0\}$. It is clear that the case $r \notin N(U)$ corresponds to facet lifting, and $r \in N(U)$ to facet routing. The lifting transforms the digraph $G = (N(U), U)$ into digraph $G_r = (N(U) \cup r, U_r)$ by adding the two arcs (r, i) and (j, r) for every arc $(i, j) \in U$. Notice that an arc (r, k) or (k, r) might be added multiple times; the aggregate multiplicity corresponds to the coefficient of the new arc in the lifted inequality. The routing transforms a digraph $G = (N(U), U)$ into digraph $G' = (N(U), U')$ by adding the two arcs (r, i) and (j, r) for every arc $(i, j) \in U :\; i, j \neq r$ and by changing the orientation of the arcs (r, j) and $(i, r) \in U$.

Remark 2.3. If $indeg(s) = outdeg(s)$ for all $s \in N(U)$, then lifting does not give new facets.

The routing transforms the digraph $G = (N(U), U)$ into digraph $G_r = (N(U), U')$ by adding two arcs (r, i) and (j, r) for every arc $(i, j) \in U : i, j \neq r$ and by changing the orientation of arcs $(r, j) \in U$ and $(i, r) \in U$.

Let $ax \leq a_0$ be a facet-defining inequality; then for any i and j, $a\psi_j(\psi_i(x)) \leq a_0$ is equivalent either to $ax \leq a_0$ or to $a\psi_k(x) \leq a_0$, for some $k \in N$. It does not allow us to apply the rotation method repeatedly.

We call a valid inequality $ax \leq a_0$ for P_n *regular*, if

$$\sum_{i\in N} a_{is} = \sum_{i\in N} a_{si} \quad \text{for all } s \in N. \tag{2.4}$$

If $a_{ij} \in \{0, 1\}$, then condition (2.4) is equivalent to $indeg(s) = outdeg(s)$ for all $s \in N(U)$.

For $r \in N(U)$ let

$$a_{ij}^r = \begin{cases} 0 & \text{if } i = r \text{ or } j = r, \\ a_{ij} & \text{otherwise,} \end{cases}$$

$$a_0^r = a_0 - \frac{1}{2}\left(\sum_{i\in N} a_{ir} + \sum_{i\in N} a_{ri} \right).$$

The next lemma is a generalization of the McLennan transitivity proposition [11].

LEMMA 2.3 (routing lemma). *If $ax \le a_0$ is a regular inequality for P_n and $r \in N(U)$, then $ax \le a_0$ represents a facet of P_n iff $a^r x \le a_0^r$ represents a facet of P_n.*

To prove the lemma it suffices to note that

$$\text{routing } \psi_r : ax \le a_0 \Rightarrow a^r x \le a_0^r \text{ and}$$
$$\text{lifting } \psi_r : a^r x \le a_0^r \Rightarrow ax \le a_0.$$

Remark 2.4. If a digraph $G = (N(U), U)$ induces a facet F and if for all $s \in N(U)$, $indeg(s) = outdeg(s)$, then deletion of any vertex s gives a digraph, which itself induces a facet.

3. Reducing forms of new facet-defining inequalities. The rotation ψ_r of a facet $\sum a_{ij}x_{ij} \le a_0$ yields the facet-defining inequality $\sum \bar{a}_{ij}x_{ij} \le \bar{a}_0$ with the following coefficients:

$$\bar{a}_{ij} = a_{ij}, i \ne r, j \ne r,$$
$$\bar{a}_{sr} = \sum_{(j,s)\in U} a_{js},$$
$$\bar{a}_{rs} = \sum_{(s,j)\in U} a_{sj},$$
$$\bar{a}_0 = a_0 + \sum_{(i,j)\in U} a_{ij} \qquad \text{(lifting)},$$
$$\bar{a}_0 = a_0 + \sum_{\substack{(i,j)\in U \\ i,j\ne r}} a_{ij} \qquad \text{(routing)}.$$

The equality $x_{ij} + x_{ji} = 1$ allows one to reduce oppositely directed arcs in the digraph G, and it corresponds to the following coefficient corrections:

$$a'_{ij} = \bar{a}_{ij} = a_{ij}, i \ne r, j \ne r, \tag{3.1}$$
$$a'_{sr} = \max\{\bar{a}_{sr} - \bar{a}_{rs}, 0\}, \tag{3.2}$$
$$a'_{rs} = \max\{\bar{a}_{rs} - \bar{a}_{sr}, 0\}, \tag{3.3}$$
$$a'_0 = \bar{a}_0 - \sum_{s\in N(U)} \min(\bar{a}_{sr}, \bar{a}_{rs}) \qquad \text{(lifting)}, \tag{3.4}$$
$$a'_0 = \bar{a}_0 - \sum_{s\in N(U)\setminus\{r\}} \min(\bar{a}_{sr}, \bar{a}_{rs}) \quad \text{(routing)}. \tag{3.5}$$

If a facet-defining inequality has the form

$$x(U) = \sum_{(i,j)\in U} x_{ij} \le a_0 \tag{3.6}$$

(all facets described above have this form), then the rotation ψ_r of (3.6) is

$$\sum_{(i,j)\in U'} x_{ij} + \sum_{s\in N(U')} (\max\{0, indeg(s) - outdeg(s)\}x_{sr}$$
$$+ \max\{0, -indeg(s) + outdeg(s)\}x_{rs}) \le a'_0,$$

where $a'_0 = a_0 + |U'| - \sum_{s\in N(U')} \min\{indeg(s), outdeg(s)\}$, $U' = \{(i,j) \in U :\ i,j \ne r\}$ and $indeg(s)(outdeg(s))$ is the number of incoming (outcoming) arcs for a node s. Remember, that $a_{rs} = 1,\ a_{sr} = 0$, if $(r,s) \in U,\ a_{rs} = 1,\ a_{rs} = 0$, if $(s,r) \in U$.

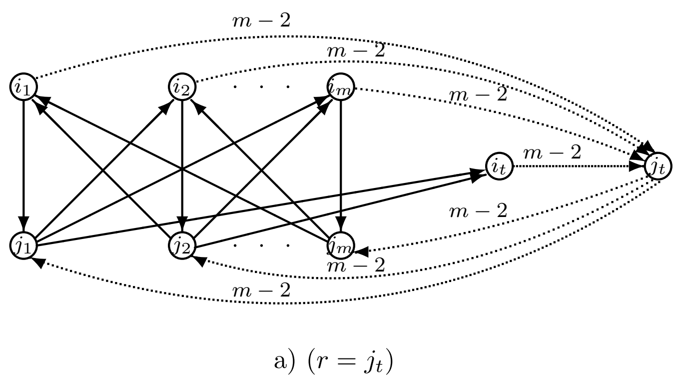
4. Rotations of m-fences. The proof of the next theorem (and of all other similar theorems in this paper) directly follows from Theorem 2.2 and (3.1)–(3.5).

THEOREM 4.1. *The following inequalities define facets of the linear ordering polytope:*
(a) *the routing of m-fences for all $r \in \{i_1, \dots, i_m, j_1, \dots, j_m\}$:*

$$\sum_{\substack{s=1 \\ i_s \neq r \ \ or \ \ j_s \neq r}}^{m} \left(x_{i_s j_s} + (m-2)(x_{i_s r} + x_{r j_s}) + \sum_{\substack{q=1 \\ q \neq s, j_q \neq r}}^{m} x_{j_q i_s} \right) \leq 2(m-1)^2;$$

(b) *the lifting of m-fences for all $r \notin \{i_1, \dots, i_m, j_1, \dots, j_m\}$:*

$$\sum_{s=1}^{m} \left(x_{i_s j_s} + (m-2)(x_{i_s r} + x_{r j_s}) + \sum_{\substack{q=1 \\ q \neq s}}^{m} x_{j_q i_s} \right) \leq 2m^2 - 3m + 1.$$



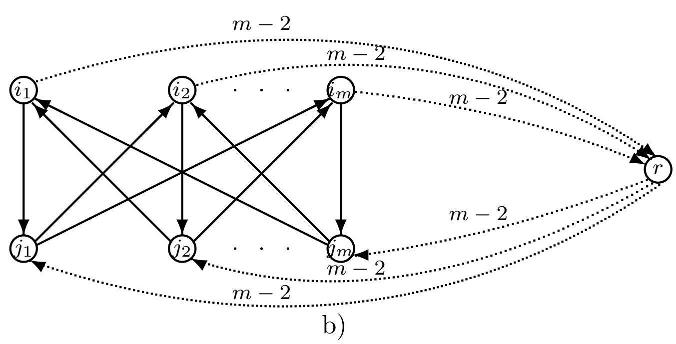


FIG. 2. *Routing* (a) *and lifting* (b) *of m-fence.*

The facets which are depicted in Figure 2(a) and 2(b) cut off noninteger vertices $x = \psi_r(x^0)$, $\psi_r(x^{00})$, respectively.

Note that the facet in Figure 2(a) for $m = 3$ is isomorphic to the Möbius ladder in Figure 4.

5. Rotations of t-reinforced m-fences. A generalization of m-fences was presented in [6, 10, 11]. If $(I \cup J, F)$ is an m-fence, then for any nonnegative integer

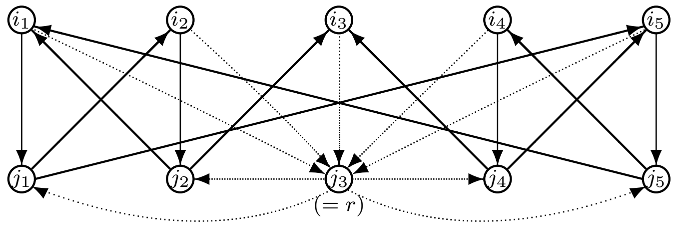


FIG. 3. *Routing of* $(5,2)$*-fence for* $r = j_3$.

$t \le m - 2$ the inequality

$$t \sum_{s=1}^{m} x_{i_s j_s} + \sum_{s=1}^{m} \sum_{\substack{q=1 \\ q \ne s}}^{m} x_{j_s i_q} \le m(m-1) + \frac{t(t+1)}{2}$$

represents a facet of P_n and is called a *t-reinforced m-fence* [10]. Application of the rotation method to t-reinforced m-fences yields new classes of facets.

THEOREM 5.1. *The following inequalities define facets of the linear ordering polytope:*
(a) *the routing of t-reinforced m-fences for any $t = 1, 2, \ldots, m-2$:*

$$\sum_{\substack{s=1 \\ i_s \ne r \text{ or } j_s \ne r}}^{m} \left(t x_{i_s j_s} + (m-t-1)(x_{i_s r} + x_{r j_s}) + \sum_{\substack{q=1 \\ q \ne s, j_q \ne r}}^{m} x_{j_q i_s} \right) \le 2(m-1)^2 + tm + \frac{t(t+1)}{2};$$

(b) *the lifting of t-reinforced m-fences for any $t = 1, 2, \ldots, m-2$:*

$$\sum_{s=1}^{m} \left(t x_{i_s j_s} + (m-1-t)(x_{i_s r} + x_{r j_s}) + \sum_{\substack{q=1 \\ q \ne s}}^{m} x_{j_q i_s} \right) \le (2m-2-t)m + \frac{t(t+1)}{2}.$$

The last class of facets coincides with *augmented t-reinforced m-fences* introduced by McLennan [11] and Leung and Lee [10].

6. Rotations of (m,k)-fences. An example of the routing of an (m,k)-fence is presented in Figure 3.

THEOREM 6.1. *The following inequalities define facets of the linear ordering polytope:*
(a) *the routing of (m,k)-fences for any $r \in \{i_1, \ldots, i_m, j_1, \ldots, j_m\}$:*

$$\sum_{\substack{s=1 \\ i_s \ne r \\ j_s \ne r}}^{m} \left(x_{i_s j_s} + (2k-3)\left(x_{i_s r} + x_{r j_s}\right) + \sum_{\substack{q=1 \\ j_{s+q} \ne r \\ j_{s-q} \ne r}}^{k-1} \left(x_{j_{s+q} i_s} + x_{j_{s-q} i_s}\right) \right)$$
$$\le \frac{m+1}{k} - 1 + 2(m-1)(k-1) + m(2k-3);$$

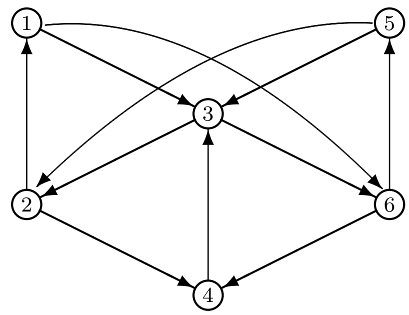


FIG. 4. *Möbius ladder, $m = 5$.*

(b) *the lifting of (m,k)-fences for any $r \notin \{i_1, \dots, i_m, j_1, \dots, j_m\}$:*

$$\sum_{s=1}^{m} \left(x_{i_s j_s} + (2k-3)(x_{i_s r} + x_{r j_s}) + \sum_{q=1}^{k-1} (x_{j_{s+q} i_s} + x_{j_{s-q} i_s}) \right) \leq \frac{m+1}{k} - 1 + 2m(k-1) + m(2k-3).$$

7. Rotations of Möbius ladders. The following class of facet-defining inequalities was constructed in [9] and [12]:

$$x(M) = \sum_{(i,j)\in M} x_{ij} \leq |M| - \frac{m+1}{2}, \tag{7.1}$$

where M is a set of arcs of the *Möbius ladder*, which, in turn, is an ordered sequence of odd number of different dicycles $C_1, \dots, C_m$ of length 3 or 4 (see Figure 4), satisfying the following condition: Every two adjacent dicycles have a common arc, and if any two of these dicycles C_i and C_j, $C_i < C_j$ have a common node, then this node belongs to either all dicycles $C_i, \dots, C_j$ or all dicycles $C_j, \dots, C_m, C_1, \dots, C_i$ (see an exact definition in [12]). Note that, for an $(m,2)$-fence, inequality (1.7) has the form

$$\sum_{s=1}^{m} (x_{i_s j_s} + x_{j_{s+1} i_s} + x_{j_{s-1} i_s}) \leq \frac{m+1}{2} + 2m - 1 \tag{7.2}$$

and coincides with the Möbius ladder for all dicycles of length 4 (4-dicycles). If $m = 3$, then inequality (7.2) coincides with inequalities (1.6) and (1.7), i.e., 3-fences, (3,2)-fences, and Möbius ladders with three 4-dicycles are isomorphic.

THEOREM 7.1. *The following inequalities define facets of the linear ordering polytope:*

(a) *the routing of Möbius ladders:*

$$\sum_{(i,j)\in M'} (x_{ij} + x_{jr} + x_{ri}) + \sum_{(i,r)\in M} x_{ri} + \sum_{(r,i)\in M} x_{ir} \leq |M| + |M'| - \frac{m+1}{2},$$

where $M' = \{(i,j) \in M \ : \ i,j \neq r\}$;

(b) *the lifting of Möbius ladders:*

$$\sum_{(i,j)\in M} (x_{ij} + x_{jr} + x_{ri}) \leq 2|M| - \frac{m+1}{2}.$$

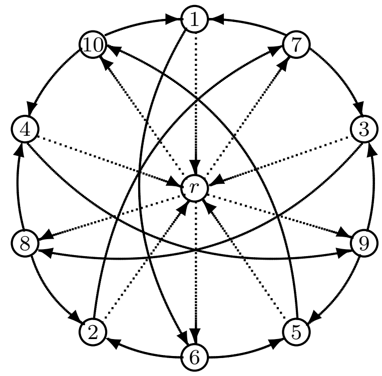


FIG. 5. *Lifting of a Möbius ladder.*

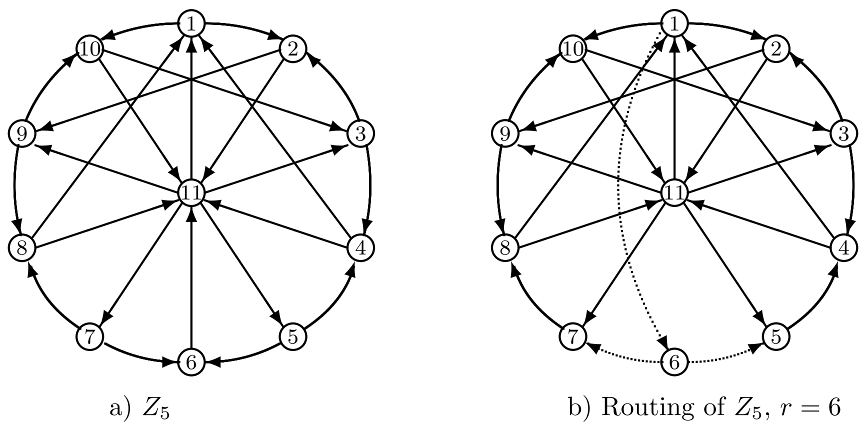


FIG. 6.

Let M consist of 4-dicycles only. Note that there exists a unique Möbius ladder (up to isomorphism) generated by m 4-dicycles. Observe that such a Möbius ladder is isomorphic to an $(m, 2)$-fence. The lifting of the Möbius ladder with every dicycle of length 4 (or, in other words, lifting of the $(m, 2)$-fence) is the facet

$$\sum_{s=1}^{m} \left(x_{i_s j_s} + x_{j_{s+1} i_s} + x_{j_{s-1} i_s} + x_{r i_s} + x_{j_s r}\right) \leq \frac{7m-1}{2}$$

named in [12] as an m-wheel with center in r. The corresponding digraph is depicted in Figure 5, where $i_s = 2s - 1,\ j_s = 2s,\ s = 1, 2, 3, 4, 5$ (cf. Figure 1(b)).

8. Rotations of $\boldsymbol{Z_m}$-facets. In [12] the facet-defining digraph $Z_m = (N, A)$ is introduced as a generalization of Möbius ladders.

Let M be the set of arcs of a Möbius ladder in which there exists only one 4-dicycle (j_1, i_m, j_m, i_2) and all 3-dicycles (i_s, j_{s-1}, h), (i_s, j_s, h), $s = 2, \ldots, m$, have a common node h. Then $A = M \cup \{(j_{m-1}, i_1), (j_2, i_1), (i_1, j_m), (i_1, j_1)\}$. See Figure 6(a) for the corresponding digraph $Z_m = (I \cup J \cup h)$, where $m = 5$, $h = 11$, $i_s = 2s - 1$, $j_s = 2s$, $s = 1, 2, \ldots, 5$.

THEOREM 8.1. *The following inequalities define facets of the linear ordering polytope:*

(a) *the routing of Z_m-facets:*

$$\sum_{(i,j)\in A'}(x_{ij}+x_{jr}+x_{ri})+\sum_{(i,r)\in A}x_{ri}+\sum_{(r,i)\in A}x_{ir}\le |A|+|A'|-m-2,$$

where $A'=\{(i,j)\in A\ :\ i,j\neq r\}$*;*

(b) *the lifting of Z_m-facets:*

$$\sum_{(i,j)\in A}(x_{ij}+x_{jr}+x_{ri})\le 2|A|-m-2.$$

It should be noted that Z_m is a facet of P_n only for $m\ge 5$, $n\ge(2m+1)$. An example of routing ψ_6 of Z_5 is depicted in Figure 6(b).

Acknowledgments. The authors would like to thank the referees for helpful suggestions which have improved the presentation of this paper.

REFERENCES

[1] G. G. Bolotashvili, *On the facets of the permutation polytope*, Commun. Georgian Acad. Sci., 121 (1986), pp. 281–284 (in Russian).

[2] G. G. Bolotashvili, *A Class of Facets of the Permutation Polytope and a Method for Constructing Facets of the Permutation Polytope*, Preprint VINITI N3403-B87, Moscow, 1987 (in Russian).

[3] G. G. Bolotashvili and M. M. Kovalev, *The partial order polytope*, in Proceedings of the VIII conference "Problems in theoretical cybernetics," Gorky, 1988 (in Russian).

[4] V. J. Bowman, *Permutation polyhedra*, SIAM J. Appl. Math., 22 (1972), pp. 586–589.

[5] P. Fishburn, *Binary probabilities induced by rankings*, SIAM J. Discrete Math., 3 (1990), pp. 478–488.

[6] P. Fishburn, *Induced binary probabilities and the linear ordering polytope: A status report*, Math. Social Sci., 23 (1992), pp. 67–80.

[7] E. Girlich, G. G. Bolotashvili, and M. M. Kovalev, *The poset polytope*, in European Chapter on Combinatorial Optimization VIII, Poznan, 1995.

[8] M. Grötschel, M. Jünger, and G. Reinelt, *A cutting plane algorithm for the linear ordering problem*, Oper. Res., 32 (1984), pp. 1195–1220.

[9] M. Grötschel, M. Jünger, and G. Reinelt, *Facets of the linear ordering polytope*, Math. Programming, 33 (1985), pp. 43–60.

[10] J. Leung and J. Lee, *More facets from fences for linear ordering and acyclic subgraph polytopes*, Discrete Appl. Math., 50 (1994), pp. 185–200.

[11] A. McLennan, *Binary stochastic choice*, in Preferences, Uncertainty and Optimality, J. S. Chipman, D. McFadden, and M. K. Richter, eds., Westview Press, Boulder, CO, 1990, pp. 187–202.

[12] G. Reinelt, *The Linear Ordering Problem: Algorithms and Applications*, Res. Exp. Math. 8, Heldermann Verlag, Berlin, 1985.

[13] G. Reinelt, *A note on small linear ordering polytope*, Discrete Comput. Geom., 10 (1993), pp. 67–78.

[14] V. A. Yemelichev, M. M. Kovalev, and M. K. Kravtsov, *Polytopes, Graphs, Optimization*, Cambridge University Press, Cambridge, UK, 1984.

SIAM J. DISCRETE MATH.
Vol. 12, No. 3, pp. 337-341

EMBEDDING GRAPHS INTO A THREE PAGE BOOK WITH $O(M \log N)$ CROSSINGS OF EDGES OVER THE SPINE*

HIKOE ENOMOTO† AND MIKI SHIMABARA MIYAUCHI‡

Abstract. This paper studies the problem of embedding a graph G into a book with vertices on a line along the spine of the book and edges on the pages in such a way that no edge crosses another. When each edge is allowed to appear in one or more pages by crossing the spine, one of the authors showed that there exists a three page book embedding of G in which each edge crosses the spine at most $O(n)$ times, where n is the number of vertices. This paper improves the result and shows that there exists a three page book embedding of G in which each edge crosses the spine at most $O(\log n)$ times. An $\Omega(n^2)$ lower bound on the number of crossings of edges over the spine in any book embedding of the complete graph K_n is also shown.

Key words. graphs, book embedding, crossings of edges over the spine

AMS subject classifications. 05C10, 05C85, 68R10

PII. S0895480195280319

1. Introduction.

A *d page book* consists of a line L (called the *spine*) in R^3 together with d distinct half-planes (called *pages*) with L as their common boundary. A *d page book embedding* of a graph G is an embedding of G into a d page book that places each vertex on the spine and draws each edge on pages so that edges on the same page do not intersect and each edge crosses only the spine; neither overlap the spine nor touch the spine; see Figure 1.1. That is, the vertices are distinct points on the spine and the edges are simple Jordan curves between them in such a way that any two curves are either disjoint or meet only at their common endpoint. Additionally, each curve can cross the spine L of the book but doesn't permit overlapping or touching L.

Bernhart and Kainen [2] and subsequently Chung, Leighton, and Rosenberg [3] defined book embeddings of a graph so as to prevent edges from crossing the spine (which we call *a combinatorial book embedding*), as shown in the left figure in Figure 2.1, and many combinatorial book embedding problems have been studied (see [5, 8]).

Another type of book embedding is possible, where each edge is allowed to appear in *one or more pages* by crossing the spine. We call this type of book embedding *a topological book embedding*. The points (excluding vertices) at which edges of G cross the spine are called *division points*, and division points and vertices are called *division vertices*. A connected part of an edge assigned to a page is called a *subedge*. In this formulation, a graph G can be embedded in a one page book if and only if G is outerplanar, and G can be embedded in a two page book if and only if G is planar. Atneosen ([1, see Corollary 2.4]) and independently Bernhart and Kainen ([2, see Theorem 5.4]) have shown the existence of a three page topological book embedding for an arbitrary graph. Miyauchi [6] examined the number of division points on the spine and found that Atneosen's proof is not constructive and that

*Received by the editors January 20, 1995; accepted for publication (in revised form) March 5, 1999; published electronically September 7, 1999.
http://www.siam.org/journals/sidma/12-3/28031.html
†Department of Mathematics, Keio University, Yokohama-Shi, Kanagawa 223-8522 Japan (enomoto@math.keio.ac.jp).
‡NTT Communication Science Laboratories, Atsugi-Shi, Kanagawa 243-0198 Japan (miyauchi@theory.brl.ntt.co.jp).

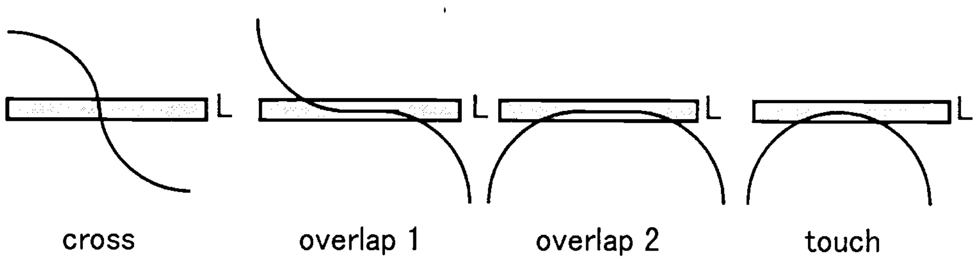


FIG. 1.1. *The way of crossing with an edge and the spine.*

Bernhart and Kainen's embedding idea needs at least $3\nu(G)$ division points, where $\nu(G)$ is the *crossing number,* i.e., the minimum number of pairwise intersections of its edges when G is drawn in a plane. In particular, for a complete graph K_n with n vertices, Bernhart and Kainen's embedding idea needs $3n^4$ division points. Miyauchi [6] also shows that there exists a three page book embedding of a graph G in which each edge crosses the spine at most n times, where n is the number of vertices. Thus it is an interesting problem to determine the minimum number of division points on the spine of a three page topological book embedding for a given graph.

This paper improves Miyauchi's result. We show that there exists a three page topological book embedding of a graph G in which each edge crosses the spine at most $2\log\ n+1$ times. To prove this theorem, first we lay out vertices of G along the spine of the book and then embed edges in pages so that conflicting subedges are embedded in different pages, where subedges (s,t) and (p,q) *conflict* if their endpoints (division vertices) are laid out in such an order as s,p,t,q or p,s,q,t on the spine from left to right.

A complete graph K_6 can be embedded in a three page book without any division points on the spine, as shown in the left figure in Figure 2.1. But a complete graph greater than K_6 cannot be embedded in a three page book without any division points. In fact, for $n \geq 7$, we show that any three page topological book embedding of K_n has at least $(n-3)(n-6)/2$ division points. Note that this result says that every topological book embedding of a complete graph K_7 with seven vertices has at least two division points on the spine of the book.

2. Three page book embeddings with $O(m\log n)$ division points. In this section, we construct a topological book embedding of an arbitrary graph $G=(V,E)$ into a three page book with $O(m\log n)$ division points on the spine, where n is the number of vertices and m is the number of edges. We may assume that G is simple; that is, G has neither loops nor multiple edges. Let $S=\{0,1\}$ be the binary alphabet and S^* the set of all strings over S. If $s\in S^*$ has length k ($k>0$), then write $s=s_1s_2\cdots s_k$, where s_i is the character of s in position i. Order the elements of S by $0<1$. Define the lexicographic order $<$ on S^* as follows. Suppose $s,t\in S^*$. If i is the first position where s and t differ and $s_i<t_i$, then $s<t$. If s and t agree in all positions that they have in common and s is shorter than t, then $s<t$. As an example, the strings of length at most 2 are ordered as follows:

$$\epsilon < 0 < 00 < 01 < 1 < 10 < 11,$$

where ϵ denotes the empty string. Define $k=\lceil \log_2 n\rceil$. A vertex s in $V(G)=\{0,\ldots,n-1\}$ has a unique representation as a string in S^k using binary representation.

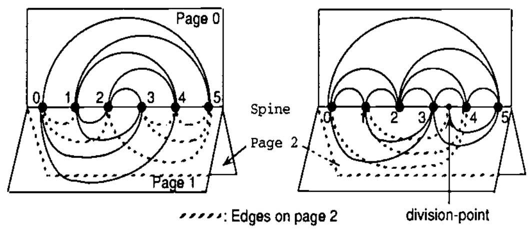


FIG. 2.1. *Three page book embeddings of* K_6.

For a vertex $s \in V(G)$, use the representation $s_1 \cdots s_k$ for its binary representation, where s_1 is the highest-order binary digit. For a string $s = s_1 \cdots s_k$ in S^k let $s(i)$ be the string consisting of the first i letters of s, that is, $s(i) = s_1 \cdots s_i$ and $s(0)$ be the empty string ϵ. For two distinct strings s and t in S^k, define $\ell = lcp(s,t)$ to be the length of the *longest common prefix* of s and t, that is, $s(\ell) = t(\ell)$ while $s_{\ell+1} \neq t_{\ell+1}$.

Consider a subdivision G^* of G which is made by subdividing each edge $(s,t) \in E(G)$ by adding vertices labeled

$$(s,t;k-1), (s,t;k-2), \ldots, (s,t;\ell) = (t,s;\ell), \ldots, (t,s;k-2), (t,s;k-1)$$

in this order from s to t, where $\ell = lcp(s,t)$, and identify $(s,t;\ell)$ with $(t,s;\ell)$. Since a combinatorial book embedding of G^* into a three page book corresponds to a topological book embedding of G by regarding vertices in $V(G^*) - V(G)$ as division points, we will construct a combinatorial three page book embedding of G^*. First, we lay out the vertices of G^* along the spine of the book and then add three numbers 0, 1, and 2 (the "pages") to edges of G^* so that conflicting edges receive different numbers.

THEOREM 2.1. *There exists a three page combinatorial book embedding of the subdivision* G^* *of a graph* G.

Proof. First, we lay out the vertices of G^* on the spine. The vertices in $V(G^*)$ are represented as follows:

$$V(G^*) = \{(s,t;i) | st \in E(G), lcp(s,t) \leq i \leq k\},$$

where $(s,t;k)$ is identified with s. For two edges (s,t), $(p,q) \in E(G)$, we denote $(s,t) <_* (p,q)$ when either of the following two conditions is satisfied:

i. $\min(s,t) < \min(p,q)$.

ii. $\min(s,t) = \min(p,q)$, $\max(s,t) < \max(p,q)$.

Two division vertices $(s,t;i)$, $(p,q;j) \in V(G^*)$ are laid out on the spine from left to right if one of the following three conditions holds:

1. $s(i) < p(j)$.
2. $s(i) = p(j)$, $i =$ even, and $(s,t) <_* (p,q)$.
3. $s(i) = p(j)$, $i =$ odd, and $(p,q) <_* (s,t)$.

For two division vertices $(s,t;i)$, $(p,q;j)$ when $(p,q;j)$ is laid out at the right of $(s,t;i)$ on the spine of the book, we denote $(s,t;i) < (p,q;j)$. Note that for any two division vertices $(s,t;i)$, $(p,q;j) \in V(G^*)$, either $(s,t;i) < (p,q;j)$ or $(p,q;j) < (s,t;i)$ holds. For example, the division vertices on the spine for a complete graph K_4 (i.e., $k = 2$) is ordered on the spine as shown in Figure 2.2.

As for the adjacency relations of vertices in $V(G^*)$, connect $(s,t;i-1)$ and $(s,t;i)$ $(lcp(s,t) < i \leq k)$. The edge $((s,t;i-1),(s,t;i))$ is embedded in page $c(s(i))$, which is defined by the following recursive definition:

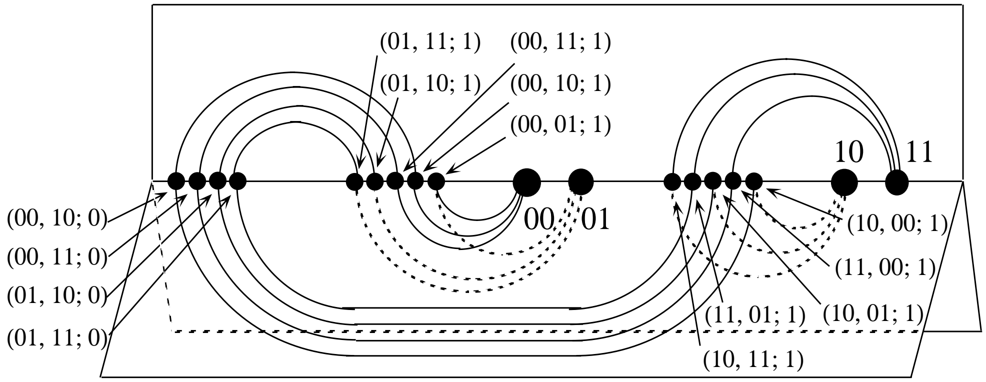


FIG. 2.2. *Division vertices on the spine of a three page book for* K_4.

1. $c(\epsilon) = 0$.
2. $c(s(i-1)s_i) = c(s(i-1)) + s_i + 1 \mod 3$.

From this recursive formula, we have $c(s(i)) = \sum_{j=1}^{i} s_j + i \bmod 3$. Note that $c(s(i-1))$, $c(s(i-1)0)$, and $c(s(i-1)1)$ are three different numbers.

Finally, we show that no two edges in $E(G^*)$ embedded in the same page conflict, i.e., the layout is legal. Let $((s,t;i-1),(s,t;i))$ and $((p,q;j-1),(p,q;j))$ be two edges in $E(G^*)$ ($lcp(s,t) < i \le k$, $lcp(p,q) < j \le k$) that conflict. Note that by the definition of the layout on the spine, $(s,t;i-1) < (s,t;i)$ and $(p,q;j-1) < (p,q;j)$. Without loss of generality, we may assume that the endpoints of the two edges appear on the spine in the order

$$(s,t;i-1) < (p,q;j-1) < (s,t;i) < (p,q;j).$$

We need to show that the two edges are embedded in different pages, that is, that $c(s(i)) \ne c(p(j))$. By the assumption and the definition of layout of vertices, $s(i-1) \le p(j-1) \le s(i) \le p(j)$. Let $\ell = lcp(s,p)$.

Case 1. $\ell < i$. In this case, since $s(i) \le p(j)$, $\ell < j$ and $s_{\ell+1} < p_{\ell+1}$. Because $p(j-1) \le s(i)$, $\ell = j-1$. Then $i = \ell + 1 = j$, because $s(i-1) \le p(j-1)$ and $\ell < i$. Thus, $c(s(i)) = c(s(\ell)) + s_i + 1 \ne c(s(\ell)) + p_j + 1 = c(P(j)) \mod 3$.

Case 2. $\ell \ge i$. In this case, since $p(j-1) \le s(i)$, $j - 1 \le i$. Because $s(i-1) \le p(j-1)$, $i \le j \le i+1$. Thus, $j = i$ or $j = i+1$. If $j = i$, then $s(i) = p(j)$ and $s(i-1) = p(j-1)$. Thus by the regulation of layout for division points, if $(s,t;i-1) < (p,q;j-1)$, then $(p,q;j) < (s,t;i)$. This contradicts the assumption $(s,t;i) < (p,q;j)$. Therefore $i \ne j$. If $j = i+1$, then $c(p(j)) = c(s(i)) + p_j + 1 \ne c(s(i))$ mod 3. ◻

In the subdivision G^*, each edge (s,t) of G is divided by adding $2(k-\ell)-1$ division points, where $\ell = lcp(s,t)$ and $k = \lceil \log_2 n \rceil$. Thus we have the following theorem.

THEOREM 2.2. *There exists a three page book embedding of a graph G in which each edge crosses the spine at most $2\log n + 1$ times, where n is the number of vertices.*

Finally, we develop a lower bound on the number of division points p on the spine in a three page topological book embedding of K_n. We first proved that $p \ge n(n-1)/56$ by using the induction on the number of the division points. Later, the following proof was suggested by Dr. K. Ota.

THEOREM 2.3. *Let K_n be a complete graph with n vertices, and let p be the number of division points on the spine of a topological three page book embedding of K_n. Then,*

$$p \geq (n-3)(n-6)/2.$$

To prove Theorem 2.3 we use the following lemma.

LEMMA 2.4 (Bernhart and Kainen [2]). *Let G be a graph with m edges and n vertices, and let $bt(G)$ be the pagenumber of any combinatorial book embedding of G. Then,*

$$bt(G) \geq (m-n)/(n-3).$$

Proof of Theorem 2.3. For a topological book embedding of K_n, let K_n^* be the subdivision of K_n whose vertices consist of vertices in $V(K_n)$ and division points. Let P be the set of division points (i.e., $P = V(K_n^*) - V(K_n)$) and let p be the number of division points (i.e., $p = |P|$). Then the number of edges in $K_n^* - P$ is at least $n(n-1)/2 - p$. Because the subgraph $K_n^* - P$ of K_n^* is embedded into a three page book without any division points, we have $(n(n-1)/2 - p - n)/(n-3) \leq 3$ by using Lemma 2.4. □

From Theorem 2.3, we found that every topological book embedding of a complete graph K_7 with seven vertices has at least two division points on the spine of the book. Furthermore, in [4] we show that the lower bound of the number of division points on the spine for a complete graph K_n with n vertices is $\Omega(n^2 \log n)$. That is, the upper bound shown in Theorem 2.2 is optimal as to the order of the number of division points.

Acknowledgments. The authors would like to thank the referees for their helpful comments and suggestions.

REFERENCES

[1] G. A. ATNEOSEN, *On the Embeddability of Compacta in n-books*, Ph.D. thesis, Michigan State Univ., East Lansing, MI, 1968.

[2] F. BERNHART AND P. C. KAINEN, *The book thickness of a graph*, J. Combin. Theory Ser. B, 27 (1979), pp. 320–331.

[3] F. R. K. CHUNG, F. T. LEIGHTON, AND A. L. ROSENBERG, *Embedding graphs in books: A layout problem with applications to VLSI design*, SIAM J. Algebraic Discrete Methods, 8 (1987), pp. 33–58.

[4] H. ENOMOTO, M. S. MIYAUCHI, AND K. OTA, *Lower bounds for the number of edge-crossings over the spine in a topological book embedding of a graph,* Discrete Appl. Math., 92 (1999), pp. 149–155.

[5] S. M. MALITZ, *Genus g graphs have pagenumber $O(\sqrt{g})$*, J. Algorithms, 17 (1994), pp. 85–109.

[6] M. S. MIYAUCHI, *An $O(mn)$ algorithm for embedding graphs into a 3-page book*, Trans. IEICE, E77-A, 3 (1994), pp. 521–526.

[7] A. T. WHITE, *Graphs, Groups and Surfaces*, North-Holland, Amsterdam, New York, 1984, p. 59.

[8] M. YANNAKAKIS, *Embedding planar graphs in four pages*, J. Comput. System Sci., 38 (1973), pp. 36–67.

SIAM J. DISCRETE MATH.
Vol. 12, No. 3, pp. 342–359

FINDING SUBSETS MAXIMIZING MINIMUM STRUCTURES*

MAGNÚS M. HALLDÓRSSON†, KAZUO IWANO‡, NAOKI KATOH§, AND TAKESHI TOKUYAMA‡

Abstract. We consider the problem of finding a set of k vertices in a graph that are in some sense *remote*. Stated more formally, given a graph G and an integer k, find a set P of k vertices for which the total weight of a *minimum structure* on P is *maximized*. In particular, we are interested in three problems of this type, where the structure to be minimized is a spanning tree (REMOTE-MST), Steiner tree, or traveling salesperson tour.

We study a natural greedy algorithm that simultaneously approximates all three problems on metric graphs. For instance, its performance ratio for REMOTE-MST is exactly 4, while this problem is NP-hard to approximate within a factor of less than 2. We also give a better approximation for graphs induced by Euclidean points in the plane, present an exact algorithm for graphs whose distances correspond to shortest-path distances in a tree, and prove hardness and approximability results for general graphs.

Key words. minimum spanning tree, traveling salesperson tour, Steiner tree, dispersion

AMS subject classifications. 58Q25, 05C85, 05C05

PII. S0895480196309791

1. Introduction.

Let $G[P]$ denote the subgraph of a graph G induced by a vertex subset P. We are interested in the following spanning tree (REMOTE-MST) problem:

> REMOTE-MST. Given a complete undirected edge-weighted graph $G = (V, E)$ and integer k, find a subset P of V of cardinality k such that the cost of the minimum weight spanning tree on $G[P]$ is maximized.

We also study the traveling salesperson (REMOTE-TSP) and Steiner tree (REMOTE-ST) problems, where the objective is to maximize the minimum traveling salesman tour and the minimum Steiner tree of the induced subgraph, respectively. These problems are illustrated in Figure 1.1.

Minimum weight spanning trees (MST), minimum weight Steiner trees, and minimum weight tours (TSP, or traveling salesperson tours) are fundamental combinatorial structures, and the problems of finding such optimal structures are not only useful in applications but also a rich source of research on exact and approximate algorithms. All of these problems consist of finding a subset *maximizing* the total weight of edges of minimum combinatorial structures constructed from the subsets. Except for REMOTE-ST, these structures are contained in the subgraph induced by the subset.

---

*Received by the editors September 27, 1996; accepted for publication (in revised form) March 1, 1999; published electronically September 7, 1999. A preliminary version of this paper appeared in *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, San Francisco, CA, 1995.

http://www.siam.org/journals/sidma/12-3/30979.html

†Science Institute, University of Iceland, IS-107 Reykjavik, Iceland (mmh@hi.is).

‡IBM Tokyo Research Laboratory, Yamato, Kanagawa 242, Japan (iwano@trl.ibm.co.jp, ttoku@trl.ibm.co.jp).

§Department of Architecture, Kyoto University, Yoshida-Honmachi, Sakyou-Ku, Kyoto 60601, Japan (naoki@is-mj.archi.kyoto-u.ac.jp). The research of this author was performed at Kobe University of Commerce.

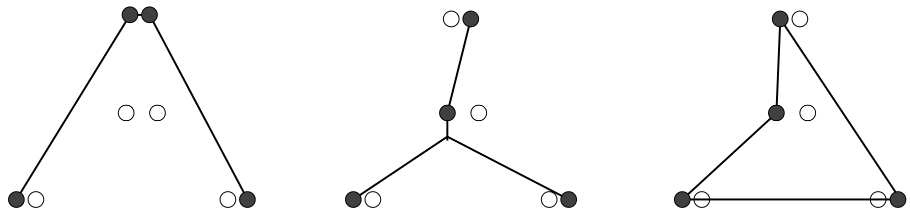


FIG. 1.1. *Remote planar point sets.*

From a practical point of view, the REMOTE-MST (or REMOTE-ST) k-set of a network can be viewed as the set of k nodes among which communicating information is most expensive. Thus, the remote subsets can be applied to the evaluation of the communication performance of networks.

They can also be applied to clustering problems. Indeed, we originally faced these problems when trying to find a good "starting tour" of a large TSP instance (a circuit board drilling problem [19] that occurred at a manufacturing plant) with more than 10,000 nonuniformly distributed cities. To obtain a short approximate TSP tour by construction heuristics, it is effective to start with a subtour (starting tour) consisting of a relatively small number of sample cities capturing the global structure of the point distribution [20]. For this purpose, random sampling is not suitable, since it may miss some critical cities, and approximate TSP tours constructed from the associated starting tour often respond poorly to improvements by local search heuristics. The exact or approximate REMOTE-MST and REMOTE-TSP solutions seem to give better starting tours.

General framework. The problems under study can be generalized to the following framework. Let Π be a minimization problem whose solution is a subset of the edge set satisfying a particular property with respect to a given subset P of vertices. Let the cost of a solution be the sum of the weight of the edges in the solution. Let $\pi(P)$ denote the minimum cost value for a node set P. We are interested in the following problem:

> REMOTE-Π. Given a graph $G = (V, E)$ and integer k, find a subset P of V of cardinality k such that $\pi(P)$ is maximized.

Our results. In section 2 we present approximation algorithms for metric graphs, general graphs, Euclidean graphs, and trees.

Metric graphs are graphs with nonnegative weights that satisfy the triangular inequality: for any three nodes u, v, w, $d(u, v) + d(v, w) \geq d(u, w)$. The *distance* of the edge (u, v), denoted $d(u, v)$, is the weight of the edge. One example of a metric graph is the *shortest-path distance graph* $D(G)$ of a graph G, where the weight of the edge (u, v) in $D(G)$ is defined to be the weight of the minimum weight path between u and v of G.

We apply in section 2.1 a known greedy algorithm to obtain simultaneous approximations of all three problems in metric graphs. We obtain performance ratios of 4 for REMOTE-MST and 3 for REMOTE-TSP, both of which are tight for this algorithm, while the ratio for REMOTE-ST is at most 3 and at least 2.46.

For REMOTE-MST in general graphs, we give in section 2.2 an algorithm that finds a solution within a factor of $k - 1$ from optimal.

Table 1.1
Approximability of remote problems.

Π	General		Metric		$\mathbf{R}^2$
	l.b.	u.b.	l.b.	u.b.	u.b.
MST	$n^{1-\epsilon}$	$k-1$	2	4	2.25
TSP	∞		2	3	
ST	4/3	3	4/3	3	2.16

Euclidean graphs are a special class of metric graphs, where the vertices correspond to points in the plane and the weight of an edge is the Euclidean distance between the points. The results obtained for the metric case, in combination with results on the *Steiner ratio* in the plane, yield asymptotic ratios of 2.31 (resp., 2.16) for the Remote-MST (resp., Remote-ST) problem.

In section 2.4, we give linear time algorithms for computing Remote-ST and Remote-MST when the set of edges in G with noninfinity weights forms a tree.

In section 3, we prove approximation hardness results for the three problems. Let n denote the number of vertices in the input graph. Remote-MST of general graphs cannot be approximated within a factor of $\Omega(n^{1-\epsilon})$, for any $\epsilon > 0$, unless $NP \subseteq ZPP$ (i.e., unless polynomially bounded zero-error randomized algorithms exist for all problems in NP). This proof is generalized to the remoteness versions of *degree-constrained subgraph* problems, with or without connectivity requirement. These problems include MST, TSP, minimum weight matching, cycle cover, degree-constrained spanning tree, and a number of other well-studied problems. For Remote-TSP we can prove a still harder inapproximability bound, since like the ordinary TSP problem, it cannot be approximated on general graphs within *any* ratio, unless $P = NP$.

On metric graphs, these problems are also NP-hard to approximate within a factor less than 2. Without loss of generality we may assume that the input to the Remote-ST problem is a metric graph. We show it to be hard to approximate within a ratio less than 4/3.

We summarize the main approximability results of the paper in Table 1.1. It lists the results obtained for each of the MST, TSP, and ST remote problems with lower and upper bounds for approximability in general graphs, metric graphs, and Euclidean graphs.

Related work. Problems of maximizing minimum structures have applications to the location of undesirable facilities. For instance, hazardous facilities like nuclear plants or ammunition dumps should be located as far from each other as possible to minimize vulnerability. A not insubstantial body of literature has been developed on the subject; see [11] for a survey, primarily from a management science viewpoint. The focus has been on two structures not dealt with in this paper: the minimum weight of any edge in the k-set and the average, or equivalently the sum, of the weights of edges between pairs in the k-set. For the former problem, known as the *k-Dispersion problem*, Ravi, Rosenkrantz, and Tayi [25] showed that the greedy furthest-point algorithm obtains a performance ratio of 2 on metric graphs, improving on a weaker bound of [29]. They also showed that approximating within a factor of less than 2 is NP-hard. Independently, Tamir [27] proved the same upper bound for the same algorithm (see also [28]).

A dispersion problem with the criteria of maximizing the *average distance* between vertices in the k-set was considered by Ravi, Rosenkrantz, and Tayi in [25], and they gave a different greedy algorithm with a ratio of 4. Hassin, Rubinstein, and

Tamir [15] gave an algorithm with a performance ratio of 2. Kortsarz and Peleg [17] considered this latter problem on general weighted graphs, under the name *heavy subgraph problem*, and gave a sequence of algorithms that converges with a performance ratio of $O(n^{0.3885})$. While different minimum structures have been proposed in the location theory literature, we are not aware of work analyzing algorithms for such problems.

Problems on Euclidean graphs can be regarded as belonging to computational geometry. The problems of finding a subset with cardinality k of a planar point set maximizing the perimeter or area of convex hull (minimum perimeter enclosing polygon) of the subset has been studied in the literature and nearly linear time algorithms are known [2, 3, 7]. However, the authors know no previous results on computing subsets maximizing other minimum structures.

Problems of finding subsets *minimizing* the minimum weight of a combinatorial structure are more common [1, 10, 24, 14]. In particular, the problem of finding the k-set minimizing the weight of the minimum MST was studied by Ravi et al. [24], who proved NP-hardness and gave the first approximations. The performance ratios have recently been improved to 3 for general graphs [14] and $1+\epsilon$ for Euclidean graphs [21, 5].

Chandra and Halldórsson [8] continued the work started in this paper and analyzed a number of other remote problems. In particular, they gave an $O(\log k)$-approximate algorithm for two problems suggested in a previous version of the current paper: computing a k-set maximizing the minimum weight matching, and the *k-defense* problem, where the objective $\pi(P)$ is $\sum_{v\in P}\min_{u\in P-\{v\}} d(u,v)$.

Notation. A *spanning tree* of a node set P is a subtree of G whose node set is P. A *Steiner tree* of P is a spanning tree of a *superset* of P. A *tour* of P is a cycle that contains all the vertices of P. The weight of a tree or a tour is the sum of the weight of the edges in it.

We denote the minimum spanning tree, minimum Steiner tree, and TSP tour of P by $MST(P)$, $ST(P)$, and $TSP(P)$, respectively. The weights of these minimum solutions are denoted by $mst(P)$, $st(P)$, and $tsp(P)$. For a graph H, the maximum cost of $MST(P)$ over all k-node sets P is denoted by r-mst(H). In general, for a problem Π and node set P, the minimum structure and the minimum value are denoted by $\Pi(P)$ and $\pi(P)$, respectively, and the optimal value of Remote-Π (i.e., the maximum weight of the minimum Π-structure) is denoted by r-$\pi(H)$.

The *approximation ratio* of an algorithm for Remote-MST on a given input graph G is the ratio of the largest MST weight of a set of k points to the MST weight of the k-set output by the algorithm. The same holds for other problems. The *performance ratio* ρ of the algorithm is the maximum approximation ratio over all instances. A problem is *approximable within a factor of* t if there exists a polynomial time algorithm for the problem with a performance ratio at most t. A problem Π_1 is *as hard to approximate* as problem Π_2 if an approximation of Π_2 within a factor of $f(n)$ implies an approximation of Π_1 within a factor of $O(f(n))$.

Given a graph G and value γ, the bivalued network $H_{G,\gamma}$ is a complete graph on the same vertex set as G, where the weight of an edge is 1 if the edge is in G and γ otherwise. Let $G[P]$ denote the subgraph of G induced by a vertex subset P. Namely, $P \subset V(G)$ and $E(G[P]) = \{(v,u) \mid (v,u) \in E(G) \text{ and } v,u \in P \subseteq V(G)\}$. The distance graph $D(G)$ of a graph G has the weight of an edge (u,v) equal to the length of the shortest path from u to v in G.

2. Algorithms.

2.1. Metric graphs. In this section, we assume that $G = (V, E)$ is metric unless otherwise stated. Let the distance between a node u and a set of nodes be the minimum distance between u and any node in the set, $d(v, P) = \min_{p \in P} d(v, p)$.

Central to our approach is the concept of an *anticover*.

DEFINITION 2.1. *A set P of vertices $p_1, p_2, \ldots$ is an anticover of a graph if*
1. $d(p_i, p_j) \geq r$ *for* $i \neq j$ *and*
2. $\min_i \{d(v, p_i)\} \leq r$ *for any node* $v \in V$.

The radius *of P is the largest value r for which P is an anticover. The* size *of an anticover is its number of vertices.*

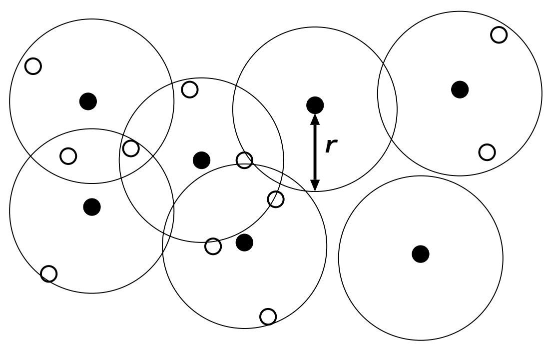


FIG. 2.1. *Anticover (black points) of size* 7 *of a Euclidean graph.*

An anticover is illustrated in Figure 2.1. An anticover can be constructed efficiently by the greedy furthest-point algorithm [12, 29, 25] given in Figure 2.2.

```
Greedy(G)
  pick an arbitrary node v
  P ← {v}
  for i ← 2 to k
     v ← node in V − P furthest from P
     P ← P ∪ {v}
end
```

FIG. 2.2. *The greedy furthest-point algorithm.*

It is easy to see that the node set found by Greedy is an anticover of size k and that its radius is the distance between the node v chosen last and $P - \{v\}$.

We apply Greedy to obtain simultaneous constant-factor approximations of the remote MST, TSP, and Steiner problems. The same algorithm was applied to approximate the k-Dispersion problem [29, 25] as well as the Euclidean k-clustering problem [12], indicating a level of universality of this approach and an applicability to multiobjective computing.

THEOREM 2.2. *An anticover of size k is a* 4*-approximation of* REMOTE-MST *and a* 3*-approximation of* REMOTE-ST *and* REMOTE-TSP.

Proof. Let P be an anticover of G and let r denote its radius. Let Q be any set of k points.

Any pair of points in P is of distance at least r, so

$$mst(P) \geq (k-1)r. \tag{2.1}$$

Each point q in Q is of distance at most r from P; thus the tree obtained by connecting Q to $MST(P)$ via the shortest edge is of weight at most $mst(P) + kr$. That is,

$$st(Q) \leq st(P \cup Q) \leq mst(P) + kr.$$

The ratio (Steiner ratio) of the weight of an MST of a set of k points to that of its Steiner tree is at most $2(k-1)/k$. It follows that

$$\frac{mst(Q)}{mst(P)} \leq 2\frac{k-1}{k}\left(1 + \frac{kr}{(k-1)r}\right) \leq 4 - \frac{2}{k}.$$

Similarly,

$$st(P) \geq \frac{k}{2}r$$

because of the Steiner ratio, and

$$st(Q) \leq st(P \cup Q) \leq st(P) + kr.$$

Hence, a performance ratio of 3 follows.

Furthermore,

$$tsp(P) \geq kr.$$

Connecting each point of Q to its nearest point in P by a pair of directed edges (with different directions), we can form a tour of $P \cup Q$ of length at most $tsp(P) + 2kr$. Thus,

$$tsp(Q) \leq tsp(P \cup Q) \leq tsp(P) + 2kr \leq 3 \cdot tsp(P). \qquad \square$$

The Steiner ratio $2(k-1)/k$ holds even if the tree is restricted to be a path; thus the results hold equally for degree-constrained versions of the problems.

While the analysis of the approximation ratio in Theorem 2.2 obtained by Greedy appears loose, it is actually asymptotically optimal for both Remote-MST and Remote-TSP. We give lower bounds on the performance of Greedy that holds for any choice of the initial starting vertex.

Theorem 2.3. *The performance ratio of* Greedy *for* Remote-MST *on metric graphs is asymptotically* 4.

Proof. We construct a family of instances for which Greedy is destined to perform poorly independent of its choice of a starting vertex.

Let G_t be an unweighted graph with vertex set $\{c, p_1, p_2, \ldots, p_t, q_1, q_2, \ldots, q_t\}$. Let $p_1, \ldots, p_t, c$ be connected into a path, and let each q_i be connected to both p_1 and p_2. G_t contains no further edges.

Let $G'_{t,z}$ be the graph formed by taking z copies of G_t with a single c vertex common to all copies (Figure 2.3). Thus we have a connected graph on $2tz + 1$ nodes. For convenience, we use notations such as p_1-vertex, p-vertex, q-vertex, and c-vertex. To force the algorithm to prefer the p-vertices, we perturb the distances between vertices as follows: the lengths $d(c, p_t)$ are stretched to $1 + 2\epsilon$ and the lengths $d(p_i, p_{i+1})$ to $1 + \epsilon$ for $i \geq 1$.

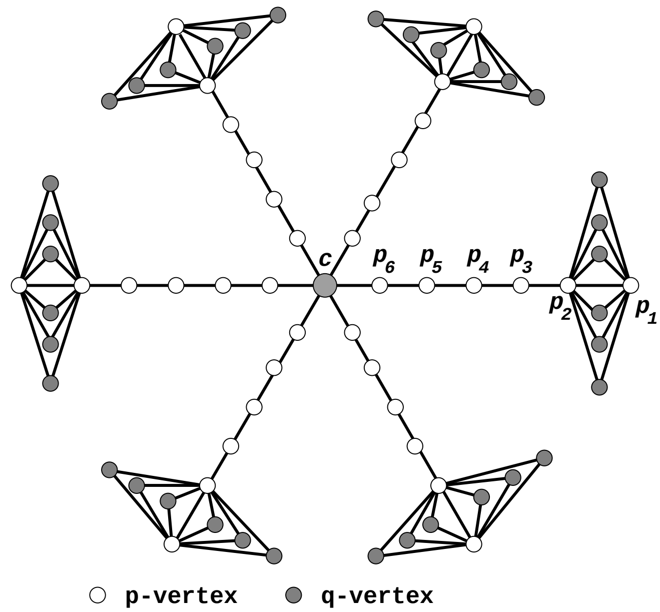


FIG. 2.3. *Lower bound example for* Greedy.

The hard instance is the distance graph $D(G'_{t,z})$ with z sufficiently large. Observe that the distance between q-vertices in different copies is $2t(1+\epsilon)$, while the distances between p_1 vertices is $2t(1+\epsilon)+2\epsilon$. Thus a p_1 vertex is the furthest vertex from any set of at most $z-1$ vertices.

Let $k = tz$. The set of the first z vertices selected by Greedy contains at least $(z-1)$ p_1-vertices. Thus, Greedy cannot select a q-vertex adjacent to a selected p_1-vertex. Consequently, the number of q-vertices which Greedy can select is at most t. Also, Greedy must select the vertex c, whose neighbors are all of distance at least $1+2\epsilon$. Thus, ignoring the ϵ terms, $mst(P) \leq zt + 2(t-1)$ for any set P of k points selected by Greedy.

Let Q consist of the tz different q-vertices. Let q_i and q'_i be vertices in different copies of G_t. Then

$$\begin{aligned} mst(Q) &= z(t-1)d(q_i, q_{i+1}) + (z-1)d(q_t, q'_1) \\ &= 2z(t-1) + (z-1)(2t) \\ &= 4zt - 2z - 2t. \end{aligned}$$

If $z = t$, we have that

$$\rho \;\geq\; \frac{mst(Q)}{mst(P)} \;=\; 4 - O\left(\frac{1}{\sqrt{k}}\right). \qquad \square$$

Although the above lower bound is applicable only to the solution generated by Greedy, we conjecture that 4, rather than the lower bound of 2 that we will give in Theorem 3.4, is the best possible performance ratio for the problem.

One plausible approach for improving on the approximation produced by Greedy is to postprocess the greedy solution with local improvement changes. Having obtained an anticover P of radius at most r, it may be possible to move individual points further away from the other points. That is, for a point $v \in P$ with $d = d(v, P - \{v\})$, there may exist a point $u \in V - P$ such that $d(u, P - \{v\}) > d$. This would improve the bounds, using a strengthening of (2.1) to $mst(P) \geq \sum_{v \in P} d(v, P - \{v\})(k-1)/k$.

The hard instances constructed above demolish that hope, since no single point can be moved further away. These instances can be easily modified to ensure that no b points can be moved further away, for any fixed b.

THEOREM 2.4. *The performance ratio of* Greedy *for* REMOTE-TSP *is asymptotically* 3.

Proof. Our construction is based on the graphs $G'_{t,z}$ of the preceding theorem. Assume z is even and consider an arbitrary matching of z copies of G_t into $z/2$ pairs. Assign the weight $\alpha = \sqrt{t}$ to each edge between each pair G_t and $G_{t'}$. Among these, we add an additional ϵ weight to the edges incident on p_1-vertices to ensure they will always be favored.

Our graph $G''_{t,z}$ is the graph obtained by adding the above edges to the original $G'_{t,z}$. Then Greedy selects the same set P as in Theorem 2.3, and there is a tour of P using edges from $MST(P)$ as well as $z/2$ matching edges between p_1 vertices. Thus $tsp(P) = zt + o(zt)$. On the other hand, $tsp(Q) \geq 3zt$ for the set Q consisting of the q-vertices. ◻

THEOREM 2.5. *The performance ratio of* Greedy *for the* REMOTE-ST *problem is at least* $32/13 \geq 2.46$.

Proof. Let $z = k/2$. Let ϵ be a number less than $z^{-1/2}$.

Let T be an edge-weighted tree with $V(T) = \{c, p, q, u_1, u_2, r\}$. Let $d(p, u_1) = d(p, u_2) = 12$, $d(p, r) = 7$, $d(c, r) = 1$, and $d(r, q) = 5 + \epsilon$ for a positive real number ϵ. We extend this tree by adding edges to obtain a complete graph T' in which the distance between any pair of vertices is the minimum of 16 and the shortest distance within this tree.

We construct a graph H containing z copies $T'(1), \dots, T'(z)$ of T'. Let the copy of a vertex v in $T'(i)$ be denoted by $v(i)$. The vertices $c(1), c(2), \dots, c(z)$ are located on a path with the distance between $c(i)$ and $c(j)$ is $|i - j| z^{-1/2}$. Define $dist_H(x, y)$ to be the minimum of 16 and the shortest path distance on H.

We extend H by adding edges to obtain a graph G whose distance function is denoted by $dist_G$. $dist_G$ equals $dist_H$ within each $T'(i)$ $(i = 1, 2, .., z)$, while for vertices $v(i)$ and $w(j)$ in distinct copies of T',

$$dist_G(v(i), w(j)) = \begin{cases} dist_H(p(i), p(j)), & v = p,\ w = p, \\ \min\{dist_G(p(i), p(j)) - \epsilon, dist_H(v(i), p(j))\}, & v \neq p,\ w = p, \\ \min\{dist_G(p(i), p(j)) - 2\epsilon, dist_H(v(i), w(j))\}, & v \neq p,\ w \neq p. \end{cases}$$

We can easily check that $dist_G$ satisfies the triangle inequality. By construction, $p(j)$ is the node in $T'(j)$ that is the farthest from $p(i)$ for any $i \neq j$. Also, if $|j - i| > 16z^{1/2}$, then $p(j)$ is the node in $T'(j)$ that is the farthest from any other node in $T'(i)$.

Greedy applied to G first selects a node, say $c(i_0)$, and then picks all $p(j)$ for which $|j - i_0| > 16z^{1/2}$. So far, the distance from the farthest node to the current vertex set is at least $16 + z^{1/2}$. Next, it picks at most two nodes (typically, $u_1(j)$ and $u_2(j)$) in each $T'(j)$ satisfying $|j - i_0| < 16z^{1/2}$.

Now, the distance from the farthest node to the current vertex set is reduced to $12 + \epsilon$, and the algorithm selects all $q(j)$ for which $|j - i_0| > 16z^{1/2}$. The algorithm has by now selected $k - O(z^{1/2})$ nodes; the choice of the remaining $O(z^{1/2})$ is irrelevant to our analysis.

The length of the spanning tree of the output of the algorithm is $(13+\epsilon)z + O(z^{1/2})$. On the other hand, if we pick $u_1(i)$ and $u_2(i)$ for $i = 1, 2, .., z$, the length of the spanning tree is $32z - O(z^{1/2})$. Hence, as z goes to infinity, the ratio between the cost of the optimal solution to that of the greedy solution approaches $32/13 \approx 2.4615$. □

The precise determination of the performance ratio of Greedy for Remote-ST remains an open problem.

2.2. General graphs. We give an approximation algorithm for Remote-MST on general graphs, with a performance ratio of $k-1$.

For a graph G and a positive weight α, define G_α to be the subgraph of G on $V(G)$ with edges whose weight is less than α.

HeavyEdge(G)
  Determine the largest α such that
    G_α is not $(n-k+1)$-vertex-connected.
  Let C be a cutset of G_α of size $n-k$.
  Output $P = V - C$.
end

The desired α can be found by binary search on the at most $\binom{n}{2}$ different edge-weights. Since P, the subgraph induced by C, is not connected in G_α, an MST of P must contain an edge of weight at least α. On the other hand, if edges of weight α are added to G_α, any k-set must be connected. Thus,

$$r\text{-}mst(G) \le (k-1)\alpha \le (k-1)mst(P).$$

Corollary 2.6. HeavyEdge *has performance ratio of* $k-1$ *for* Remote-MST.

For the Steiner tree problem, it suffices to consider the distance graph of the input graph, which satisfies the triangular inequality. Thus we obtain the following corollary of Theorem 2.2.

Corollary 2.7. Remote-ST *of a general graph can be approximated within a factor of* 3.

2.3. Euclidean graphs. Let P be a set of n points $\{p_1, \ldots, p_n\}$ in the plane. The Euclidean graph of P is the complete graph on the node set P, where the weight of an edge (p_i, p_j) is the Euclidean distance $d(p_i, p_j)$. We consider algorithms for approximating Remote-MST and Remote-ST of this graph.

The anticover defined in the previous section gives a geometric covering of P by k circles of radius r, each of which is centered by a point in P. Since $st(P) \ge \sqrt{3}mst(P)/2$ [9] in the Euclidean case, we immediately obtain the following.

Corollary 2.8. *An anticover is a* $\frac{4k-2}{\sqrt{3}(k-1)}$*-approximation of* Remote-MST *and a* $\frac{2k+\sqrt{3}(k-1)}{\sqrt{3}(k-1)}$*-approximation of* Remote-ST *in Euclidean graphs.*

Thus, the approximation ratios are asymptotically at most $4/\sqrt{3} \approx 2.309$ for Remote-MST and $(2+\sqrt{3})/\sqrt{3} \approx 2.155$ for Remote-ST.

Unlike in the metric case, it seems that the approximation ratio depends on the choice of the anticover. For the example in Figure 2.4, the worst anticover has a $(2\sqrt{3}+4)/3) \approx 2.448$ approximation ratio, which is near to the upper bound $14/3\sqrt{3} \approx 2.694$ for the Remote-MST 4-set.

2.4. Tree networks. In this section, we consider graphs in which the set of edges with finite weights forms a tree. Let T be a weighted tree on n nodes. Define

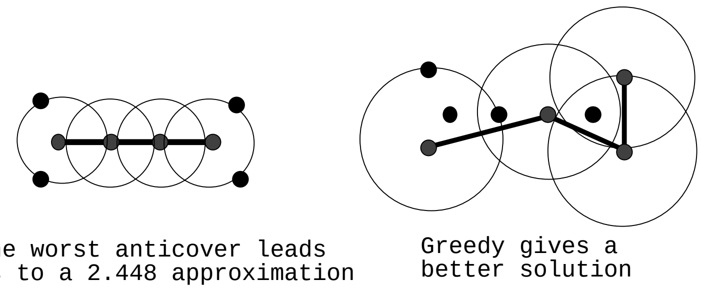


FIG. 2.4. *Approximation by circle covers.*

$G(T)$ to be a complete graph of order n, where the weight of an edge is the same as in T if it exists in T and ∞ otherwise.

We first give efficient algorithms for the REMOTE-ST k-set of $G(T)$. Clearly, we should select k leaves. If $k = 2$, the problem is the diameter path problem on a tree—finding the pair of vertices of maximum distance—and can be solved in linear time. If k exceeds the number of leaves of T, every set of k nodes containing all leaves forms the (unique) optimal REMOTE-ST k-set. The following lemma (essentially given in Peng, Stephens, and Yesha [23] for a slightly different problem) is the key observation.

LEMMA 2.9. *Any optimal* REMOTE-ST $(k-1)$*-set is contained in an optimal* REMOTE-ST *k-set.*

Proof. Let S_1 be an optimal REMOTE-ST $(k-1)$-set and let S_2 be any optimal REMOTE-ST k-set. Let T_1 (T_2) be the Steiner tree spanned by S_1 (S_2) and let $W(T_1)$ ($W(T_2)$) be its weight. Then $T_1 \cap T_2$ is a (possibly empty) tree.

The edge sets $F_1 = T_1 - T_1 \cap T_2$ and $F_2 = T_2 - T_1 \cap T_2$ are forests. Each connected component (a tree) in the forest has a root, which is a vertex in $T_1 \cap T_2$. If $T_1 \cap T_2$ is empty, we pick arbitrary vertices x in T_1 and y in T_2, consider the path from x to y in T, and define the root of T_1 (resp., T_2) as its nearest node to y (resp., x) on the path.

A leaf of these forests must be in either $S_2 - S_1$ or $S_1 - S_2$. For each leaf v of these forests, let $h(v)$ be the weight of the path from v to the root in the component T_0 containing v, and $l(v)$ be the weight of the path to the nearest branch from v in T_0 (to the root if there is no branch). Then $h(v) \geq l(v)$, and equality holds if and only if T_0 is a path.

Consider a leaf v in $S_1 - S_2$ and a leaf w in $S_2 - S_1$. Unless $h(v) = l(v) = h(w) = l(w)$, either $h(w) > l(v)$ or $h(v) > l(w)$ holds. If $h(w) > l(v)$, we consider the set $S_1 - \{v\} + \{w\}$ and observe that the spanning tree of this set has the weight $W(T_1) + h(w) - l(v) > W(T_1)$. This contradicts the assumption that S_1 is an optimal REMOTE-ST $(k-1)$-set. Similarly, if $h(v) > l(w)$, we derive a contradiction to the fact that S_2 is an optimal REMOTE-ST k-set. Therefore, $h(v) = l(v) = h(w) = l(w)$ holds for all pairs of leaves v in F_1 and w in F_2. We write l for $l(v)$.

Hence the total weight of T_1 is $W(T_1 \cap T_2) + |S_1 - S_2|l$ and that of T_2 is $W(T_1 \cap T_2) + |S_2 - S_1|l$. By definition, $|S_2 - S_1| = |S_1 - S_2| + 1$. Therefore, if we choose any $w \in S_2 - S_1$, the Steiner tree of $S_1 \cup \{w\}$ has the same length as that of S_2; hence $S_1 \cup \{w\}$ is a REMOTE-ST k-set and it contains S_1. Thus we obtain the lemma. □

This enables us to compute a REMOTE-ST k-set (more precisely, its spanning tree) by a greedy algorithm: Starting from any diameter path, find the leaf farthest from the current tree and update the tree by adding the leaf and the path from the leaf. This takes $O(kn)$ time. We can give a better algorithm, which is an analogue of an algorithm of Shioura and Uno [26] for the problem in [23].

Consider a subtree H of T. Then $T-H$ forms a forest F of rooted, directed trees with roots drawn from the vertices of H. We partition the edges of F into a set of directed paths such that for any internal node w on a path leading to a leaf l, the weight of the subpath from w to l is maximum over all directed paths from w to a leaf. This is easy to compute bottom-up in linear time for each tree in T by selecting for each internal node the path of maximum weight through a child. Define for each leaf l in F the *benefit* of l to be the weight of the path incident on l.

In particular, we consider a diameter path P as H. The above process computes all the benefits of leaves of $T-P$ in linear time. Then we can observe the following lemma.

LEMMA 2.10. *The greedy algorithm adds the $k-2$ leaves with the largest benefits of $T-P$ to P to obtain a* REMOTE-ST *k-set.*

Proof. For any *RST* j-set and its Steiner tree H, the leaf v in $T-H$ of largest benefit is the one that is farthest from H. Moreover, for the tree H' obtained by joining the path from v to H, the benefits of $T-H'$ are by construction the same as the benefits of $T-H$ (except, of course, for v). Thus we have the lemma. ∎

We can select the k largest benefits using a linear time selection algorithm, and hence we have the following theorem.

THEOREM 2.11. *The* REMOTE-ST *k-set of $G(T)$ can be computed in $O(n)$ time.*

We next consider the REMOTE-MST problem on trees. REMOTE-MST of $G(T)$ is not a well-defined problem since we can almost always find a subset P whose MST has infinity weight. Instead, we add a *connectivity condition* to the definition of a remote k-set P, such that $mst(P)$ is maximized on the condition that $mst(P) \neq \infty$. Namely, $MST(P)$ must be an induced subtree of P in T. Thus, the problem becomes a special case (where all edge weights are nonpositive) of the *minimum weighted* $(k-1)$-*cardinality tree* problem defined by Fischetti et al. [13] if we reverse the sign of all weights of T. We can thus apply their $O(k^2n)$ time dynamic programming algorithm. Moreover, we can improve it to $O(kn)$ time.

THEOREM 2.12. *The minimum weighted $(k-1)$-cardinality tree of a weighted tree can be computed in $O(kn)$ time. Hence, the* REMOTE-MST *k-set of $G(T)$ under the connectivity condition can be computed in $O(kn)$ time.*

Proof. We give a proof only for the computation of an optimal REMOTE-MST k-set. For a node v, a *v-optimal* REMOTE-MST *j-set* refers to a j-set that induces an optimal REMOTE-MST among those j-sets constrained to contain v.

Fix any internal node r as the root of T. The *profile* of a subtree T_0 with root r_0 is the set consisting of the weights of r_0-optimal REMOTE-MST j-sets for $j = 1, 2, \ldots, \min(k, |T_0|)$, and the weight of the optimal REMOTE-MST k-set of T_0 if $k > |T_0|$. We give a dynamic programming algorithm that sweeps the tree bottom-up to compute the profile of T.

If r has only one child r_1, then the r-optimal j-set of T is the union of r with the r_1-optimal $j-1$-set of T_1. Otherwise, let r_1 be any child of r rooting the subtree T_1 and let T_2 be the tree obtained by cutting T_1 and the adjoining edge from T.

The minimum spanning tree of the optimal REMOTE-MST j-set containing r in T can be obtained by joining the r_1-optimal REMOTE-MST j_1-set of T_1 and the r_2-optimal REMOTE-MST j_2-set of T_2 for a suitable pair j_1 and j_2 satisfying $j_1 + j_2 = k$. The weight of the tree is simply the sum of these two parts. Thus the weight of the

r-optimal REMOTE-MST j-set for $j = 1, 2, \ldots k$ can be computed in $O(k^2)$ time by examining all combinations.

We can improve the analysis of the time complexity. We say that a node u of T is *heavy* if both of its descendant trees have at least $k/2$ nodes and *light* otherwise. The number of heavy nodes is at most n/k. We separately charge for operations at the heavy nodes, which is of cost $O(kn)$ in total. Let $f(n)$ be the cost for operations at all light nodes.

At a light node r rooting T with subtrees T_1 and T_2, T_i has n_i nodes and $m_i = n_i - 1$ edges. Since r is a branching node, $m_i \geq 1$ $(i = 1, 2)$. T has $m = n - 1$ edges; thus $m \geq m_1 + m_2$ holds.

The profile for T_i has $\min(n_i, k)$ weights of r_i-optimal REMOTE-MST sets. Hence to compute the profile of T, we need only to examine $\min(n_1, k)\min(n_2, k)$ combinations, which takes $O(\min(n_1, k)\min(n_2, k)) = O(\min(m_1, k)\min(m_2, k))$ time. Thus, the cost function $f(m)$ (up to a constant factor) follows the formula $f(m) \leq f(m_1) + f(m_2) + \min(m_1, k)\min(m_2, k)$.

Consider $g(m) = \min\{2km, m^2\}$. We shall verify that $g(m)$ satisfies $g(m) \geq g(m_1) + g(m_2) + \min(m_1, k)\min(m_2, k)$. Assume without loss of generality that $m_2 \leq m_1$. Thus, m_2 is smaller than $k/2$ at a light node.

Case 1. If $2k \geq m$, then $g(m) = m^2 \geq (m_1 + m_2)^2 \geq m_1^2 + m_2^2 + m_1 m_2$.

Case 2. If $m \geq 2k$, then $g(m) = 2km \geq 2km_1 + km_2 + km_2 > \min(2k, m_1)m_1 + m_2^2 + km_2$.

Hence $f(m) < cg(m)$ for some constant c and thus is $O(km)$. Since $m = n - 1$, the complexity is $O(kn)$. □

The same algorithm can compute REMOTE-MST k-sets (with connectivity condition) of decomposable graphs, such as series-parallel graphs, in $O(kn)$ time.

3. Hardness. The decision version of REMOTE-MST (to decide whether there exists a set of k vertices whose MST weight is more than a given threshold) is obviously in NP. Instead of showing NP-hardness, we show approximation-hardness for both general and metric graphs.

We shall be primarily interested in approximating the remote problems within a function independent of k. Thus we ask about the worst-case performance ratio as k ranges from 1 through n. Let $\alpha(G)$ denote the independence number of G, or the size of a maximum independent set.

THEOREM 3.1. *Approximating* REMOTE-MST *is as hard as approximating* INDEPENDENT SET.

Proof. Let g be the gap in the approximability of INDEPENDENT SET. Thus, for some value R, determining if $\alpha(G) = R$ or $\alpha(G) \leq R/g$ is hard.

Let k be R and let γ be a value greater than k. We construct a bivalued graph $H = H_{G,\gamma}$ on the same vertex set as G with the weight of an edge being 1 if contained in G and γ otherwise. Refer to Figure 3.1.

If there is an independent set of size k in G, then that set has a value $r\text{-}mst = (k-1)\gamma$. On the other hand, suppose $r\text{-}mst(H) \geq (k-1)\gamma/g$. Notice that this is at least $(k/g - 1)\gamma + (k - k/g)$, since $\gamma \geq k$. Then there is a subset P of k vertices such that $MST(P)$ contains at least $k/g - 1$ edges of weight γ. Let $G[P]$ be the subgraph in G induced by P. It follows that $G[P]$ must contain at least k/g connected components. Hence, $\alpha(G) \geq \alpha(G[P]) \geq k/g$.

It follows that

$$\alpha(G) = k \Rightarrow r\text{-}mst(H) = (k-1)\gamma,$$

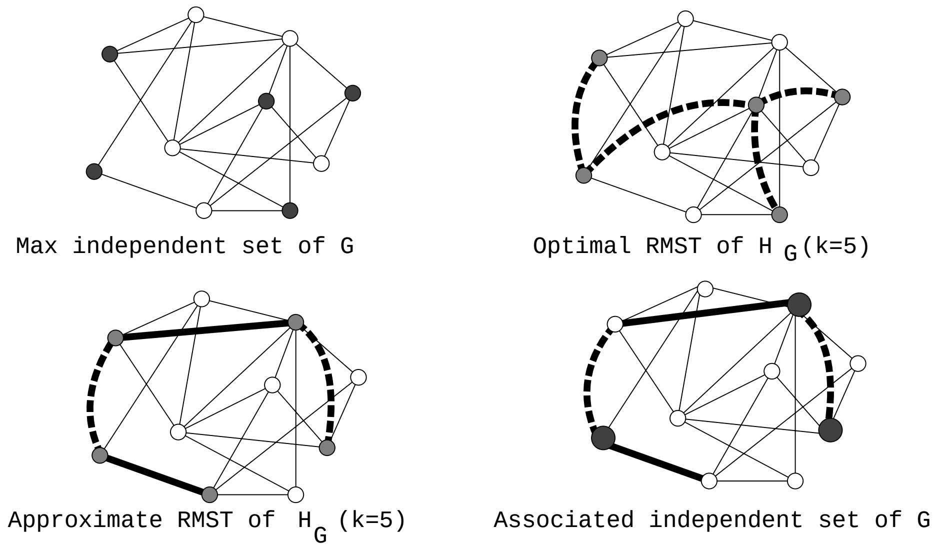


FIG. 3.1. *Graphs in Theorem* 3.1.

$$\alpha(G) \leq k/g \Rightarrow r\text{-}mst(H) \leq (k-1)\gamma/g.$$

Thus, a gap in the approximation of INDEPENDENT SET carries over to REMOTE-MST. ∎

Håstad has recently strengthened the approximation hardness of INDEPENDENT SET to $n^{1-\epsilon}$ for any $\epsilon > 0$ [16]. This assumes that $NP \not\subseteq ZPP$ or that randomized polynomial-time algorithms do not exist for NP.

We now generalize the hardness proof for REMOTE-MST to other problems. Given a graph and integers ℓ and u, the *degree-constrained subgraph problem* (DCS) is to find a subgraph of minimum weight such that the degree of each vertex is between ℓ and u, inclusive. Note that u may be only the trivial bound of $n-1$. The DCS minimization problem can be solved via a reduction to nonbipartite matching [18], and it subsumes the assignment problem and problems of covering the vertices with cycles or paths. If the subgraph must additionally be connected, we have *connected-DCS (CDCS) problems*, which include TSP, MST, and the degree-constrained MST problems.

Assume hereafter that Π is any such DCS problem with degree lower bound ℓ. For a given $\gamma > 1$, let $H = H_{G,\gamma}$ be the bivalued network on G that has the weight of an edge being 1 if the edge is in G and γ otherwise. Fix some optimal Π-solution to H and let $\Pi(H)$ denote its set of edges. We sometimes abuse notation by denoting Π for $\Pi(H)$.

LEMMA 3.2. *Let G be a graph and $\gamma \geq 1$. Let Heavy denote the set of γ-weight edges in $\Pi(H)$. Then,*

$$\alpha(G) \geq \frac{|Heavy|}{\ell^2 + 1}.$$

Proof. The statement is trivial if $|Heavy| \leq \ell^2 + 1$; thus we assume the contrary. Also, we assume that the number k of vertices in $\Pi(H)$ is greater than any constant power of ℓ.

Let *Conn* denote some minimal set of edges from *Heavy* such that $Conn \cup (\Pi(H) - Heavy)$ is connected and spans H, if Π is a problem requiring connectivity. Otherwise, let *Conn* be the empty set. Let $Slack = Heavy - Conn$. Let S denote the set of vertices incident on fewer than ℓ edges in $\Pi(H) - Slack$.

We first observe that

$$\alpha(G) \geq |Conn| + 1, \tag{3.1}$$

since G must contain that many connected components. To satisfy the lemma, it now suffices to bound the independence number in terms of the other heavy edges, by $\alpha(G) \geq |Slack|/\ell^2$.

Observe that each edge in *Slack* has an endpoint in S, and, furthermore, it has an endpoint in S that is of degree exactly ℓ in $\Pi(H)$. Otherwise, this edge would be superfluous to $\Pi(H)$, as connectivity and degree requirements are satisfied without it. This implies that

$$|Slack| \leq \ell |S|.$$

To complete the lemma, we need to show that all the edges in $G[S]$ must be in $\Pi(H)$. That implies that each vertex in S is incident on at most $\ell - 1$ edges in $G[S]$, and any maximal independent set of $G[S]$ is of size at least $\frac{1}{\ell}|S|$. Hence the independence number of the whole graph is no less, and

$$\alpha(G) \geq \alpha(G[S]) \geq \frac{1}{\ell}|S| \geq \frac{|Slack|}{\ell^2} \tag{3.2}$$

as desired.

CLAIM 1. $E(G[S]) \subseteq \Pi(H)$.

Suppose on the contrary that there were vertices x, y in S such that $(x, y) \in G$ but $(x, y) \notin \Pi(H)$. Let (x, x'), (y, y') be edges from *Slack* (where x' and y' are not necessarily distinct).

We consider three cases depending on the degrees of x' and y' (in $\Pi(H)$). If x' and y' are distinct and both of degree greater than ℓ, then let $\Pi' = (\Pi - \{(x, x'), (y, y')\}) \cup \{(x, y)\}$. If one of x' and y', say, x', is of degree greater than ℓ, then let $\Pi' = (\Pi - \{(x, x'), (y, y')\}) \cup \{(x, y), (y', z)\}$, where z is some vertex of degree ℓ nonadjacent to y'. (Such a vertex must exist since there must be at least $\ell + 1$ vertices of degree ℓ.)

Otherwise, the number of heavy edges in Π such that either x' or y' is either incident on the edge or adjacent (via an edge in $\Pi(H)$) to one of its endpoints is at most ℓ^2. Thus there must exist a third edge (x'', y'') from *Slack* such that x' and x'' are nonadjacent, as well as y' and y''. Let $\Pi' = (\Pi - \{(x, x'), (y, y'), (x'', y'')\}) \cup \{(x, y), (x', x''), (y', y'')\}$.

In all cases, the edges removed from Π are from *Slack*, and thus Π' is connected and degree constraints are preserved. Hence Π' is a valid solution of lesser cost, contradicting the minimality of Π. The claim and the lemma then follow. ∎

THEOREM 3.3. *Approximating* REMOTE-DCS *and* REMOTE-CONNECTED-DCS *problems is as hard as approximating* INDEPENDENT SET*, for any fixed value of* ℓ.

Proof. Let γ be a number greater than uk and let $H = H_{G,\gamma}$.

If there is an independent set of size k in G, then r-$\pi(H) \geq \frac{\ell}{2}k\gamma$.

On the other hand, suppose r-$\pi(H) \geq \frac{\ell}{2}k\gamma/g$. Then there is a subset P of k vertices such that $\Pi(P)$ contains at least $z \geq \frac{\ell}{2}k/g$ edges of weight γ. By Lemma 3.2, $\alpha(G[P]) \geq z/(\ell(\ell+1)) \geq k/(2(\ell+1)g) = \frac{\ell}{2}k\gamma/g'$, where $g' = g/(\ell(\ell+1))$.

It follows that

$$\alpha(G) = k \Rightarrow \text{r-}\pi(H) = \tfrac{\ell}{2}k\gamma,$$
$$\alpha(G) \le k/g \Rightarrow \text{r-}\pi(H) \le \tfrac{\ell}{2}k\gamma/g'. \qquad \square$$

Similarly, these problems are also hard to approximate in metric graphs within a factor of $2-\delta$ for any $\delta > 0$. We prove this here only for properties for which all feasible solutions have the same number of edges; the general case is quite tedious, especially for other connected properties.

THEOREM 3.4. *Let* Π *be a DCS problem with* $\ell = u$ *(e.g., TSP) or a connected property with* $\ell = 1$ *(i.e.,* (DEGREE-CONSTRAINED) MST*). Then* REMOTE-Π *is hard to approximate within a factor of* $2 - o(1)$ *in the metric space with distances* 1 *and* 2*.*

Proof. Let $\gamma = 2$, $H = H_{G,\gamma}$. Observe that any feasible solution to Π has the same number e of edges: $\ell k/2$ in the former case and $k-1$ in the latter case.

If there is an independent set of size k in G, then r-$\pi(H) = 2e$. On the other hand, suppose r-$\pi(H) \ge e(1+\delta)$. Then there is a subset P of k vertices such that $\pi(P) \ge e(1+\delta)$. Thus $\Pi(P)$ contains at least $e\delta$ edges of weight 2. By Lemma 3.2,

$$\alpha(G) \ge \frac{e\delta}{\ell(\ell+1)}.$$

Let $\delta' = \delta k/[(k-1)\ell(\ell+1)]$. Then

$$\alpha(G) = k \Rightarrow \text{r-}\pi(H) = 2e,$$
$$\alpha(G) < \delta' k \Rightarrow \text{r-}\pi(H) < e(1+\delta).$$

Hence the problem is hard to approximate within $2 - 1/f(n)$, where $f(n)$ is a function growing with n. $\square$

Theorem 3.3 can also be extended to problems involving t-connectivity (for $t = k^{o(1)}$). It can also be extended to other remote-Π problems that satisfy the following property: If F is a feasible solution to Π and (v,u) and (x,y) are edges in that solution, then $F - \{(v,u),(x,y)\} \cup \{(v,x),(u,y)\}$ is also a feasible solution to Π.

One example is when $\pi(P) = \sum_{v \in P} \min_{u \in P} d(u,v)$. The corresponding remote problem, that of finding a k-vertex set P maximizing this quantity, was considered by Moon and Chaudhry [22] under the name *k-Defense problem.* The above reduction shows that approximating it within $n^{1-\epsilon}$ in general graphs is hard.

The REMOTE-TSP problem is harder yet; like the underlying TSP problem, it cannot be approximated within any representable function.

THEOREM 3.5. *Let* W *be a polynomial representable value. Then approximating* REMOTE-TSP *in general graphs within a factor* W *is* NP*-complete.*

Proof. We give a reduction from Hamilton circuit for $k = n$.

Given a graph G on n vertices, construct the complete weighted graph H with vertex set $V(G) \cup \{u_1, \ldots, u_n\}$. Define the edge weights by

$$w(u,v) = \begin{cases} (Wn)^2 & \text{if } u,v \in V(G),\ (u,v) \notin E(G), \\ W & \text{if } u,v \in V(H) - V(G), \text{ and} \\ 1 & \text{otherwise.} \end{cases}$$

Observe that H can be represented in size polynomial in n.

Suppose G does not contain a Hamilton circuit. Then $V(G)$ is a remote k-set whose minimum salestour is of cost at least $(Wn)^2$. On the other hand, if G does contain a Hamilton circuit, then $V(H) - V(G)$ is a k-set whose minimum salestour is of cost Wn, while any other k-set has lesser cost. It follows that an algorithm that can approximate REMOTE-TSP within a factor less than Wn will decide the Hamilton circuit problem. ◻

For REMOTE-ST, one can always assume that the graph G is metric, since the minimum Steiner tree of a node set P in G can be realized in the shortest-path distance graph $D(G)$.

THEOREM 3.6. *Approximating* REMOTE-ST *within a factor of* $4/3-\delta$ *is* NP*-hard for any* $\delta > 0$.

Proof. Given a graph $G = (V, E)$, we construct a graph H as follows. Replace each edge of G by a path with two edges, and connect the middle vertices of the paths into a clique. More formally, H contains a vertex for each vertex v_i in V as well as each edge e_j in E. A vertex v_i is adjacent only to those vertices e_j for which v_i intersects e_j in G. Vertices e_j are completely connected into a clique.

The input to REMOTE-ST is the distance graph $D(H)$ of H. If we consider two vertices in G, they will be of distance 2 in H if they are adjacent in G and of distance 3 in H if they are nonadjacent in G.

An independent set in G corresponds to a set of vertices in H that have no neighbors in common. Hence, the cost of the minimum Steiner tree of that set in $D(H)$ is $2(k-1)$.

A *loner* in a Steiner tree is a leaf whose neighbor is not adjacent to another leaf. Suppose there are two loners in a Steiner tree of $D(G)$ that were adjacent in G. Then the four edges connecting them to the remaining tree could be replaced by three edges all incident on the corresponding edge-vertex in $D(G)$. Hence, given a k-set P, we can easily find a Steiner tree of P where loners form an independent set in G. If p is the number of loners, then the cost of the Steiner tree constructed will be at most $\frac{3}{2}(k-p-1) + 2p = \frac{3}{2}(k-1) + \frac{1}{2}p$.

If, now, we could guarantee finding a k-set where the minimum Steiner tree is of size at least $\frac{3}{2}k + \frac{1}{2}p$, it follows that the independence number of G is at least p. By the hardness of the independent set problem, it is hard to decide whether $r\text{-}st(G)$ is $2(k-1)$ or $(\frac{3}{2} + o(1))(k-1)$. ◻

4. Concluding remarks. If we remove the cardinality condition from the REMOTE-MST problem, we have the following problem:

> REMOTE-MST SUBSET. Find a subset Q of V such that $mst(Q)$ is maximized.

The REMOTE-MST subset problem can be considered to be an *inverse problem* to the Steiner problem. Whereas the Steiner problem asks for a superset Q' of P minimizing $MST(Q')$, the REMOTE-MST subset problem calls for a subset Q of V maximizing $MST(Q)$.

In the metric case, returning V as the solution trivially gives an approximation equal to the Steiner ratio, or 2 for general metric graphs and $2/\sqrt{3}$ for Euclidean graphs. We pose the question of improved ratios as an open problem.

Another open problem concerns the complexity classification of REMOTE-ST and REMOTE-TSP. They are in Σ_p^2, at the second level of the polynomial time hierarchy, and are NP-hard, from our results. We conjecture that they are also hard for Σ_p^2.

Other open problems include proving NP-hardness of REMOTE-MST (and perhaps MAX-SNP-hardness) in the Euclidean plane and giving better bounds for the approximation ratios for each problem. In particular, a good approximation algorithm for REMOTE-ST will be very useful in applications. Also, a fast algorithm would be needed; when we apply approximate REMOTE-TSP k-sets to large-scale TSP heuristics, subquadratic time algorithm is essential.

REFERENCES

[1] A. AGGARWAL, H. IMAI, N. KATOH, AND S. SURI, *Finding k points with minimum diameter and related problems*, J. Algorithms, 12 (1991), pp. 38–56.

[2] A. AGGARWAL, M. KLAWE, S. MORAN, P. SHOR, AND R. WILBER, *Geometric applications of a matrix-searching algorithm*, Algorithmica, 2 (1987), pp. 195–208.

[3] A. AGGARWAL, B. SCHIEBER, AND T. TOKUYAMA, *Finding a minimum weight K-link path in graphs with monge property and applications*, Discrete Comput. Geom., 12 (1994), pp. 263–280.

[4] S. ARORA, C. LUND, R. MOTWANI, M. SUDAN, AND M. SZEGEDY, *Proof verification and the hardness of approximation problems*, J. ACM, 45 (1998), pp. 501–555.

[5] S. ARORA, *Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems*, J. ACM, 45 (1998), pp. 753–783.

[6] Y. ASAHIRO, K. IWAMA, H. TAMAKI, AND T. TOKUYAMA, *Greedily finding a dense subgraph*, in Proceedings, Fifth Scandinavian Workshop on Algorithm Theory, 1996, Lecture Notes in Comput. Sci. 1097, Springer-Verlag, New York, pp. 136-145.

[7] J. E. BOYCE, D. P. DOBKIN, R. L. DRYSDALE, III, AND L. J. GUIBAS, *Finding extremal polygons*, SIAM J. Comput., 14 (1985), pp. 134–147.

[8] B. CHANDRA AND M. HALLDÓRSSON, *Facility dispersion and remote subgraphs*, in Proceedings, Fifth Scandinavian Workshop on Algorithm Theory, 1996, Lecture Notes in Comput. Sci. 1097, Springer-Verlag, New York, pp. 53–65.

[9] D.-Z. DU AND F. K. HWANG, *Proof of the Gilbert-Pollak conjecture on the Steiner ratio*, Algorithmica, 7 (1992), pp. 121–135.

[10] D. EPPSTEIN, *New algorithms for minimum area k-gons*, in Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms, Orlando, FL, 1992, pp. 83–88.

[11] E. ERKUT AND S. NEUMAN, *Analytical models for locating undesirable facilities*, European J. Oper. Res, 40 (1989), pp. 275–291.

[12] T. FEDER AND D. H. GREENE, *Optimal algorithms for approximate clustering*, in Proceedings of the 20th ACM Symposium on the Theory of Computing, Chicago, 1988, pp. 434–444.

[13] M. FISCHETTI, H. W. HAMACHER, K. JØRNSTEN, AND F. MAFFIOLI, *Weighted K-cardinality trees: Complexity and polyhedral structure*, Networks, 24 (1994), pp. 11–21.

[14] N. GARG, *A 3-approximation of the minimum tree spanning k vertices*, in Proceedings of the 37th IEEE Foundations of Computer Science, Burlington, VT, 1996, pp. 302–308.

[15] R. HASSIN, S. RUBINSTEIN, AND A. TAMIR, *Approximation algorithms for maximum facility dispersion*, Oper. Res. Lett., 21 (1997), pp. 133-137.

[16] J. HÅSTAD, *Clique is hard to approximate within $n^{1-\epsilon}$*, in Proceedings of the 37th IEEE Foundations of Computer Science, Burlington, VT, 1996, pp. 627–636.

[17] G. KORTSARZ AND D. PELEG, *On choosing a dense subgraph*, in Proceedings of the 34th IEEE Foundations of Computer Science, Palo Alto, CA, 1993, pp. 692–701.

[18] E. LAWLER, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.

[19] S. MISONO AND K. IWANO, *Circuit Board Drilling Problem*, Technical Report 93-AL-33, Inform. Process. Soc. Japan, 1993, pp. 95–102.

[20] S. MISONO AND K. IWANO, *Experiments on TSP Real Instances*, Research Report RT0153, IBM Tokyo Research Laboratory, 1996.

[21] J. S. B. MITCHELL, *Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems,* SIAM J. Comput., 28 (1999), pp. 1298–1309.

[22] I. D. MOON AND S. S. CHAUDHRY, *An analysis of network location problems with distance constraints*, Manage. Sci., 30 (1984), pp. 290–307.

[23] S. PENG, A. B. STEPHENS, AND Y. YESHA, *Algorithms for a core and k-tree core of a tree*, J. Algorithms, 15 (1993), pp. 143–159.

[24] R. RAVI, R. SUNDARAM, M. V. MARATHE, D. J. ROSENKRANTZ, AND S. S. RAVI, *Spanning trees—short or small*, SIAM J. Discrete Math, 9 (1996), pp. 178–200.

[25] S. S. RAVI, D. J. ROSENKRANTZ, AND G. K. TAYI, *Facility dispersion problems: Heuristics*

and special cases, Oper. Res., 42 (1994), pp. 299–310.

[26] A. Shioura and T. Uno, *A linear time algorithm for finding a k-tree core*, J. Algorithms, 23 (1997), pp. 281–290.

[27] A. Tamir, *Obnoxious facility location on graphs*, SIAM J. Discrete Math., 4 (1991), pp. 550–567.

[28] A. Tamir, *Comments on the paper "Facility dispersion problems: Heuristics and special cases, by S.S. Ravi, D.J. Rosenkrantz, and G.K. Tayi,"* Oper. Res., 46 (1998).

[29] D. J. White, *The maximal dispersion problem and the "first point outside the neighborhood" heuristic*, Comput. Oper. Res., 18 (1991), pp. 43–50.

SIAM J. DISCRETE MATH.
Vol. 12, No. 3, pp. 360–373

THE LINEAR EXTENSION DIAMETER OF A POSET*

STEFAN FELSNER† AND KLAUS REUTER‡

Abstract. The *distance* between two permutations of the same set X is the number of pairs of elements that are in different order in the two permutations. Given a poset $P = (X, \leq)$, a pair L_1, L_2 of linear extensions is called a *diametral pair* if it maximizes the distance among all pairs of linear extensions of P. The maximal distance is called the *linear extension diameter* of P and is denoted $led(P)$. Alternatively $led(P)$ is the maximum number of incomparable pairs of a two-dimensional extension of P. In the first part of the paper we discuss upper and lower bounds for $led(P)$. These bounds relate $led(P)$ to well-studied parameters like dimension and height. We prove that $led(P)$ is a comparability invariant and determine the linear extension diameter for the class of generalized crowns. For the Boolean lattices we have partial results.

A diametral pair generates a minimal two-dimensional extension of P or, equivalently, a maximal interval in the graph of linear extensions of P. Studies of such intervals lead to the definition of new classes of linear extensions. We give three characterizations of the class of *extremal linear extensions* which contains the greedy linear extensions. With *complementary linear extensions* we introduce a class contained in the set of super-greedy linear extensions. The complementary linear extension of L is the linear extension L^* obtained by taking the reverse of L as a priority list in the generic algorithm for linear extensions. A *complementary pair* is a pair L, M of linear extensions with $M = L^*$ and $L = M^*$. Iterations of the complementary mapping starting from an arbitrary linear extension eventually lead to a complementary pair.

Key words. poset, linear extension, diameter, greedy

AMS subject classifications. 06A07, 05C12

PII. S0895480197326139

1. Introduction and alternate formulations.

The *distance* between permutations π, σ of the same set X, denoted $dist(\pi, \sigma)$, is the number of pairs of elements that are in different order in the two permutations. Given a poset $P = (X, \leq)$, a pair L_1, L_2 of linear extensions is called a *diametral pair* if it maximizes the distance among all pairs of linear extensions of P. The maximal distance will be called the *linear extension diameter* of P and is denoted $led(P)$. In [Reu96b] the *linear extension graph* $G(P)$ was defined as the graph with vertices that are the linear extensions of P and two vertices connected by an edge if the linear extensions differ by an adjacent transposition only. Figure 1 shows the six-element poset called chevron and its linear extension graph. An easy fact about $G(P)$ is that any pair L_1, L_2 of linear extensions is connected in $G(P)$ by a path whose length equals the distance between L_1 and L_2. Hence, $led(P)$ is exactly the graph diameter of the linear extension graph $G(P)$.

The intersection of a collection $A = \{L_1, \ldots, L_k\}$ of linear extensions of P is a poset P_A which is an extension of P. The graph $G(P_A)$ is an induced subgraph of $G(P)$. Interestingly, subgraphs of $G(P)$ corresponding to extensions of P are exactly the convex subgraphs of $G(P)$ (see [BW91] or [Reu96b]).

Let $inc(P)$ denote the number of incomparable pairs of P. If L_1, L_2 is a diametral pair for P, then $P_{\{L_1,L_2\}}$ is a two-dimensional extension of P and L_1, L_2 is a diametral

*Received by the editors August 18, 1997; accepted for publication March 30, 1999; published electronically September 7, 1999.

http://www.siam.org/journals/sidma/12-3/32613.html

†Freie Universität Berlin, Fachbereich Mathematik und Informatik, Takustr. 9, 14195 Berlin, Germany (felsner@inf.fu-berlin.de).

‡Universität Hamburg, Mathematisches Seminar, Bundesstrasse 55, 20146 Hamburg, Germany (reuter@math.uni-hamburg.de).

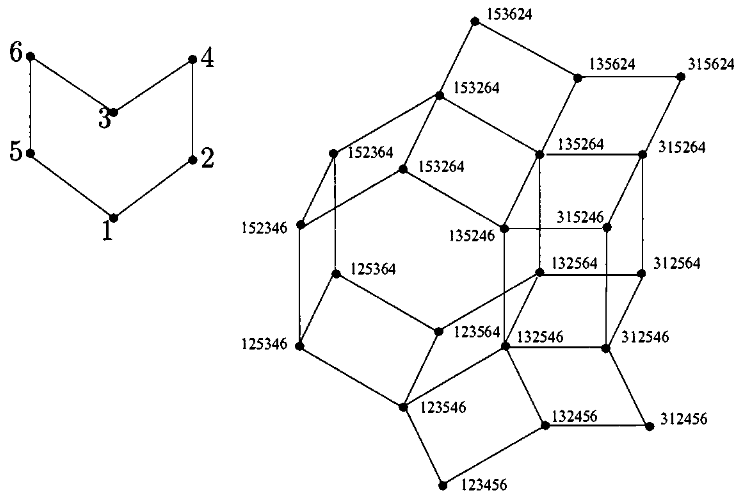


FIG. 1. *The chevron and its linear extension graph. This poset has linear extension diameter* 6.

pair for $P_{\{L_1,L_2\}}$; i.e., $led(P_{\{L_1,L_2\}}) = led(P)$. The incomparable pairs of $P_{\{L_1,L_2\}}$ are exactly the pairs being in different order in L_1 and L_2; therefore, $led(P_{\{L_1,L_2\}}) = inc(P_{\{L_1,L_2\}}) = dist(L_1, L_2)$, where $inc(P)$ denotes the number of incomparable pairs of P.

We call a two-dimensional extension Q of P a *minimum two-dimensional extension* of P if Q has a minimal number of comparable pairs that are incomparable in P. Dually, a minimum two-dimensional extension maximizes $inc(P_{\{L_1,L_2\}})$. Together with the previous paragraph this proves the following theorem.

THEOREM 1.1. *The linear extension diameter of P equals the number of incomparable pairs of a minimum two-dimensional extension of P.*

By definition $inc(Q) \leq inc(P)$ for every extension Q of P. As a consequence of the theorem we have the general bound

$$led(P) \leq inc(P). \tag{1}$$

Equality in inequality (1) is a characterization of two-dimensional posets.

THEOREM 1.2. *For a poset P the following two statements are equivalent:*

$$dim(P) \leq 2 \quad \text{and} \quad led(P) = inc(P).$$

Proof. We have already seen that $led(P) = inc(P)$ for two-dimensional posets. If P is one-dimensional, then $led(P) = 0 = inc(P)$.

For the converse suppose $led(P) = inc(P)$ and let L_1, L_2 be a diametral pair. The number of pairs being in different order in L_1 and L_2 is $inc(P)$. Therefore, P is the intersection of L_1 and L_2 which proves $dim(P) \leq 2$. □

Inequality (1) is sharp only for two-dimensional posets but, as shown with the standard examples, the following inequality may be sharp in any dimension:

$$led(P) \leq inc(P) - (dim(P) - 2). \tag{2}$$

Proof. Take a diametral pair L_1, L_2 and add one-by-one linear extensions such that $\bigcap_{i=1}^{j+1} L_i \subsetneqq \bigcap_{i=1}^{j} L_i$ until $\{L_1, \ldots, L_k\}$ is a realizer of P. Since $k \geq dim(P)$ and each L_j contributes a new incomparability to the intersection, the poset $P_{\{L_1,L_2\}}$ has at most $inc(P) - (dim(P) - 2)$ incomparable pairs. □

In the next section we give several lower bounds on the linear extension diameter. These bounds relate the new parameter to width, dimension, and fractional dimension of the poset. In section 3 we investigate the effect of small changes at the poset on its linear extension diameter. We also show that *led* is a comparability invariant. In section 4 we deal with special classes of posets. In particular, we determine the linear extension diameter of generalized crowns. Section 5 introduces the concept of complementary linear extensions as a heuristic for finding pairs of linear extensions of large distance. We prove some properties of complementary linear extensions that seem to be interesting in their own right.

2. Lower bounds on the linear extension diameter. Given a poset $P = (X, \leq)$ and disjoint subsets $A, B \subset X$, we say A *is over* B and write A/B in a linear extension L if $a > b$ in L for all incomparable pairs $a||b$ with $a \in A$ and $b \in B$. It is well known (see, e.g., [Tro92, p. 19]) that for every chain C there exist linear extensions with $C/(X \setminus C)$ and $(X \setminus C)/C$. Such a pair of linear extensions has distance at least $\sum_{x \in C} inc(x)$, where $inc(x)$ denotes the number of elements incomparable to x. Generalizing notation by defining $inc(C) = \sum_{x \in C} inc(x)$ for every chain C we have proven our first lower bound

$$\max_{C \text{ chain}} inc(C) \leq led(P). \tag{3}$$

Equality holds for the chevron and for all width-two posets. The value of this lower bound is easily computable by a maximum weighted chain algorithm. Consider a chain partition $C_1, \ldots, C_w$ of P, where $w = width(P)$. Obviously $width(P)(\max_C inc(C)) \geq \sum_{i=1}^{w} inc(C_i) = 2inc(P)$. Hence our upper and lower bounds on *led* in (1) and (3) are apart only by a factor depending on the width of P,

$$\left\lceil \frac{2inc(P)}{width(P)} \right\rceil \leq led(P) \leq inc(P). \tag{4}$$

Another lower bound relates the linear extension diameter to the dimension $dim(P)$. Take a realizer $R = \{L_1, \ldots, L_d\}$ with $d = dim(P)$ for P. Choose at random a pair S_1, S_2 of different linear extensions from R; the probability that an incomparable pair $x||y$ is incomparable in $S_1 \cap S_2$ is at least $(d-1)/\binom{d}{2}$. Therefore, the expected number of incomparable pairs in $S_1 \cap S_2$ is at least $2inc(P)/d$. This proves the bound

$$\left\lceil \frac{2inc(P)}{dim(P)} \right\rceil \leq led(P). \tag{5}$$

Since $dim(P) \leq width(P)$ this bound, (5) implies (4). Brightwell and Scheinerman [BS92] introduced the *fractional dimension* of a poset ($fdim(P)$) as the least rational number d_f such that there is an m and a multiset realizer $M = \{L_1, \ldots, L_m\}$ of P such that for every incomparable pair x, y we have $x < y$ in L_i for at least m/d_f of the linear extensions. If we choose at random a pair S_1, S_2 of linear extensions from M the probability that an incomparable pair $x||y$ is incomparable in $S_1 \cap S_2$ is at least $m/d_f(m - (m/d_f))/\binom{m}{2} = 2m(d_f - 1)/((m-1)d_f^2) \geq 2(d_f - 1)/(d_f^2)$. Since fractional dimension can be substantially smaller than the dimension, the next bound seems worthy to be stated

$$\left\lceil \frac{2(fdim(P) - 1)inc(P)}{fdim(P)^2} \right\rceil \leq led(P). \tag{6}$$

A class of orders where dimension and fractional dimension get far apart are the interval orders. The dimension of interval orders grows unbounded (see, e.g., [Tro92]) but the fractional dimension is bounded by 4 (see [BS92]). In fact, as shown recently by Trotter and Winkler [TW96], the fractional dimension of interval orders can be arbitrarily close to 4. From the above bound we thus obtain that $led(I) \geq (3/8)inc(I)$ for every interval order I. However, we can easily do better. It was shown by Rabinovich ([Tro92, p. 196]) that an interval order $I = (X, \leq)$ has a linear extension with $A/(X \setminus A)$ for every subset A of X. Choose a random subset A of X and consider two linear extensions with $A/(X \setminus A)$ and $(X \setminus A)/A$. The expected number of incomparabilities in the intersection of the two linear extensions is at least $(1/2)inc(I)$. Hence for every interval order I

$$(1/2)inc(I) \leq led(I). \tag{7}$$

The next bound relates $inc(P)$ and the height $h = height(P)$. Let $A_1, \ldots, A_h$ be the canonical antichain partition of P, i.e., $A_1 = \textsc{Min}(P)$ and for $i = 2, .., h$ $A_i = \textsc{Min}(P \setminus (A_1 \cup .. \cup A_{i-1})$. The weak order W with A_i as ith level is a two-dimensional extension of P. If $a_i = |A_i|$ the number of incomparabilities of W is $\sum_i \binom{a_i}{2}$ which is at least $h\binom{n/h}{2}$, hence, $led(P) \geq n(n-h)/2h$. For $inc(P)$ we have the obvious bound $inc(P) \leq \binom{n}{2} - \binom{h}{2}$. Therefore $inc(P) \leq n^2/2 - h^2/2 = n^2 - (1/2)(n^2 + h^2) \leq n^2 - nh$. Comparing the two inequalities we obtain

$$\lceil \frac{inc(P)}{2height(P)} \rceil \leq led(P). \tag{8}$$

The bounds of this section compare $led(P)$ to certain fractions of $inc(P)$. Graham Brightwell suggested a family P_n of random posets showing that the gap between $inc(P)$ an $led(P)$ can indeed be large. Formally, $led(P_n) = o(1)inc(P_n)$.

3. Removals and substitutions. Consider the removal of a point x from P. Let L_1, L_2 be a diametral pair for $P - x$; there exist linear extensions L_i' of P such that removing x gives L_i for $i = 1, 2$. The distance of L_1', L_2' is at least as large as the distance of L_1 and L_2; hence $led(P - x) \leq led(P)$. For a lower bound on $led(P - x)$ consider a two-dimensional extension Q of P such that $inc(Q) = led(P)$. $Q - x$ is a two-dimensional extension of $P - x$ and the incomparabilities of Q are those of $Q - x$ plus those containing element x. The incomparabilities of Q containing x are at most as many as the incomparabilities of P containing x, i.e., $inc(x)$. Hence, $led(P - x) + inc(x) \geq led(P)$.

THEOREM 3.1. *$led(P) \geq led(P - x) \geq led(P) - inc(x)$ and both inequalities can be sharp.*

Proof. It remains to show that equality may occur. Equality on both sides happens if $inc(x) = 0$. However, there are less trivial examples. On the left side take as x one of the minimal elements of **C** or **D** (these are posets from the list of 3-irreducible posets (see, e.g., [Tro92, p. 62]); **D** is the chevron). On the right side equality is attained for every two-dimensional P. ∎

Abusing the notation we write $P - r$ for the poset resulting from P after the removal of a single covering relation r. $P - r$ has more linear extensions than P; more precisely, $G(P)$ is a subgraph of $G(P - r)$. Hence, $led(P) \leq led(P - r)$. Equality is again possible: let P be the chevron augmented by the comparability $r = (1 < 3)$ (see Figure 1). A lower bound for $led(P - r)$ can be obtained from the lower bound for point removal: let r be a relation involving x, then $led(P) \geq led(P - x) = led((P - r) - x) \geq$

$led(P-r)-(inc(x)+1)$. The example of the crown $\mathbf{A}_n$ shows (see section 4) that removing r can increase *led* by as much as $(1/2)(inc(x)+1)$.

THEOREM 3.2. *Let $r=(x<y)$ be a covering relation of P; then $led(P) \leq led(P-r) \leq led(P)+\min(inc(x), inc(y))+1$.*

Let $P=(X, \leq_P)$ and $Q=(Y, \leq_Q)$ be posets on disjoint sets. Standard constructions are the parallel composition $P+Q=(X \cup Y, \leq_P \cup \leq_Q)$ and the series composition $P * Q=(X \cup Y, \leq_P \cup \leq_Q \cup(X \times Y))$. In both cases the *led* of the composition is easily determined by the components.

- $led(P+Q)=led(P)+led(Q)+|X||Y|$.
- $led(P * Q)=led(P)+led(Q)$.

Let x be an element of P and let P_x^Q be the poset obtained by substituting Q for x in P. To be more specific, $P_x^Q=((X-x) \cup Y, \leq)$ with $a \leq b$ iff $a, b \in X-x$ and $a \leq_P b$ or $a, b \in Y$ and $a \leq_Q b$ or $a \in X-x, b \in Y$ and $a \leq_P x$ or $a \in Y, b \in X-x$ and $x \leq_P b$.

THEOREM 3.3. *$led(P)+led(Q)+(led(P)-led(P-x))(|Q|-1) \leq led(P_x^Q) \leq led(P)+led(Q)+inc(x)(|Q|-1)$.*

Proof. Let L_1, L_2 be a diametral pair for P and N_1, N_2 be a diametral pair for Q. Consider the linear extensions $(L_1)_x^{N_1}$ and $(L_2)_x^{N_2}$. Compute the distance between $(L_1)_x^{N_1}$ and $(L_2)_x^{N_2}$ as the number of adjacent transpositions necessary to change $(L_1)_x^{N_1}$ into $(L_2)_x^{N_2}$ and note that changing L_1 into L_2 requires at least $led(P)-led(P-x)$ adjacent transpositions involving element x. This leads to the lower bound on $led(P_x^Q)$.

For the upper bound select an element $y \in Y$ and count the incomparabilities of a two-dimensional extension of $led(P_x^Q)$ in three parts. There are at most $led(P)$ incomparabilities between two elements in $X-x+y$, there are at most $led(Q)$ incomparabilities between two elements in Y, and finally, there are at most $inc(x)(|Q|-1)$ incomparabilities between elements of $X-x$ and elements of $Y-y$. □

Another interesting aspect of *led* is the question of comparability invariance. Reuter [Reu96a] observed that the linear extension graph $G(P)$ is not a comparability invariant. Nevertheless, as will be shown next, the linear extension diameter is a comparability invariant. The proof is based on the following lemma.

LEMMA 3.4. *The linear extension diameter of P_x^Q is attained by a pair L_1, L_2 of linear extensions in both of which the elements of Q appear consecutively.*

Proof. Let L_1, L_2 be a diametral pair of P_x^Q. Let $Q=(Y, \leq_Q)$ and choose $y \in Y$ such that in $P_{\{L_1, L_2\}}$ element y is incomparable to the maximal number of elements $z \notin Y$. Let L_1' be obtained from L_1 by first removing the elements of Y from L_1 and then reinserting them at the original position of y so that their internal order remains unchanged. Let L_2' be obtained from L_2 by the same procedure. From the choice of y it follows that the distance of L_1' and L_2' is at least as large as the distance of L_1 and L_2. Therefore, L_1', L_2' is a diametral pair and the elements of Q appear consecutively in L_1' and in L_2'. □

THEOREM 3.5. *Linear extension diameter is a comparability invariant.*

Proof. A consequence of Gallai's work [Gal67], made explicit in [DPW85], is a simple scheme for proving the comparability invariance of a property. It has to be shown only that for all posets P and Q and elements x of P, the property is unable to distinguish between P_x^Q and $P_x^{Q^d}$, where Q^d denotes the dual of Q; i.e., $y \leq y'$ in Q^d iff $y' \leq y$ in Q.

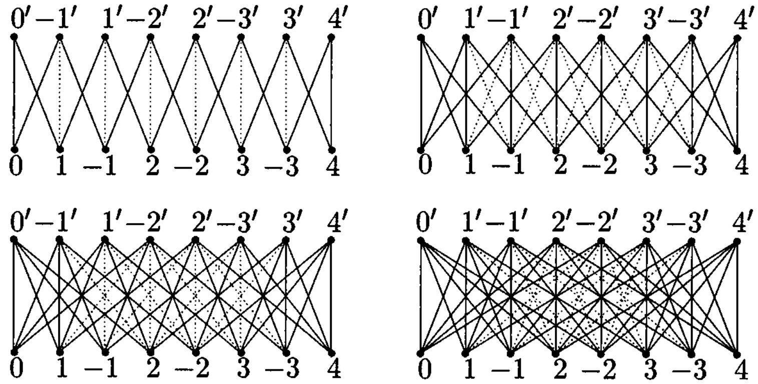


FIG. 2. *Drawings of the generalized crowns* $\mathbf{C}_8^2, \mathbf{C}_8^3, \mathbf{C}_8^4$, *and* $\mathbf{C}_8^5$. *Dotted lines indicate comparabilities of minimum two-dimensional extensions.*

Given a linear extension of P_x^Q in which the elements of Q appear consecutively we obtain a linear extension of $P_x^{Q^d}$ by reversing the order of the elements of Q. Hence, if L_1, L_2 is a diametral pair of linear extensions of P_x^Q, as in Lemma 3.4, we obtain a pair attaining the same distance for $P_x^{Q^d}$. Since the converse also works, the linear extension diameters of P_x^Q and $P_x^{Q^d}$ are equal. □

4. Generalized crowns and Boolean lattices. In this section we first deal with a class of posets where we can determine the linear extension diameter exactly. Trotter defines generalized crowns as a class of posets that interpolates between the 3-irreducible crowns $\mathbf{A}_n$ and the standard examples $\mathbf{S}_n$. For $n \geq k \geq 2$ define $\mathbf{C}_n^k$ as the height-two poset with minimal elements $\{0, 1, \dots, (n-1)\}$ and maximal elements $\{0', 1', \dots, (n-1)'\}$. Element i' is larger than the elements $\{i - \lfloor (k-1)/2 \rfloor, i - \lfloor (k-1)/2 \rfloor + 1, \dots, i + \lfloor k/2 \rfloor\}$ where indices are taken modulo n.

Lemma 4.1 can be found in [Tro92, p. 35], for the translation note that $\mathbf{C}_n^k$ equals Trotter's $\mathbf{S}_{k+1}^{n-k-1}$. In particular, $\mathbf{C}_n^2 = \mathbf{A}_n$, $\mathbf{C}_n^{n-1} = \mathbf{S}_n$, and $\mathbf{C}_n^k$ is k regular.

LEMMA 4.1. *A linear extension L of a generalized crown $\mathbf{C}_n^k$ can have $i' < j$ in L for at most $\binom{n-k+1}{2}$ pairs (i', j).*

Consider a pair L_1, L_2 of linear extensions of $\mathbf{C}_n^k$. Since each linear extension is reversing at most $\binom{n-k+1}{2}$ of the (i', j) pairs, the poset $P_{\{L_1, L_2\}}$ has at most $(n-k+1)(n-k)$ incomparable pairs $i' || j$. Adding the min/min and the max/max pairs we obtain $(n-k+1)(n-k) + n(n-1)$ as an upper bound on $led(\mathbf{C}_n^k)$. This upper bound can be attained. To simplify the exposition we assume that the minimal elements of $\mathbf{C}_n^k$ are the first n elements of the sequence $0, 1, -1, 2, -2, \dots$; the maximal elements are the corresponding primed numbers. For L_1 take the minimal elements of $\mathbf{C}_n^k$ in the order $0, 1, -1, 2, -2, \dots$ and sort in the maximal elements as early as possible. When all minimal elements have been used there are k maximal elements left; depending on the parity of k, we have taken the maximal elements in the order $0', 1', -1', 2', \dots$ (k odd) or in the order $0', -1', 1', -2', \dots$ (k even). Continue this pattern for the remaining maximal elements. For L_2 begin with the reverse ordering on the minimal elements and again sort in the maximal elements as early as possible. The final k maximal elements are taken in the reverse of their order in L_1. Figure 2 illustrates the drawings of generalized crowns resulting from this process.

Remark. A nice way of visualizing the construction is to use the diametral linear extensions as the row and column indices for the bipartite adjacency matrix of the $\mathbf{C}_n^k$. The results for $\mathbf{C}_n^3$ and $\mathbf{C}_n^4$ are displayed next. An entry $*$ at position (i, j') indicates that $i||j'$ in the crown but $i < j'$ in the two-dimensional extension:

$$
\begin{array}{c}
 \\ 0 \\ 1 \\ -1 \\ 2 \\ -2 \\ 3 \\ \vdots
\end{array}
\begin{array}{c}
\begin{array}{cccccccc} 0' & 1' & -1' & 2' & -2' & 3' & -3' & \dots \end{array} \\
\left(\begin{array}{cccccccc}
1 & 1 & 1 & & & & & \dots \\
1 & 1 & * & 1 & & & & \dots \\
1 & * & 1 & * & 1 & & & \dots \\
 & 1 & * & 1 & * & 1 & & \dots \\
 & & 1 & * & 1 & * & 1 & \dots \\
 & & & 1 & * & 1 & * & \dots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{array}\right)
\end{array}
\quad
\begin{array}{c}
 \\ 0 \\ 1 \\ -1 \\ 2 \\ -2 \\ 3 \\ \vdots
\end{array}
\begin{array}{c}
\begin{array}{cccccccc} 0' & -1' & 1' & -2' & 2' & -3' & 3' & \dots \end{array} \\
\left(\begin{array}{cccccccc}
1 & 1 & 1 & 1 & & & & \dots \\
1 & 1 & 1 & * & 1 & & & \dots \\
1 & 1 & * & 1 & * & 1 & & \dots \\
1 & * & 1 & * & 1 & * & 1 & \dots \\
 & 1 & * & 1 & * & 1 & * & \dots \\
 & & 1 & * & 1 & * & 1 & \dots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{array}\right)
\end{array}.
$$

THEOREM 4.2. *For each $n \geq k \geq 2$ the linear extension diameter of the generalized crown $\mathbf{C}_n^k$ is given by*

$$led(\mathbf{C}_n^k) = 2n(n-k) + k(k-1).$$

Proof. We have shown that $(n-k+1)(n-k)+n(n-1) = 2n(n-k)+k(k-1)$ is an upper bound on $led(\mathbf{C}_n^k)$. As for the lower bound, we have described a pair L_1, L_2 of linear extensions. From the above matrices it is easy to see that these two linear extensions have distance $(n-k+1)(n-k)+n(n-1)$. ∎

COROLLARY 4.3. *For the crown $\mathbf{A}_n$ and the standard example $\mathbf{S}_n$ this gives*

- $led(\mathbf{A}_n) = 2(n-1)^2 = inc(\mathbf{A}_n) - (n-2)$ *and*
- $led(\mathbf{S}_n) = n^2 - (n-2) = inc(\mathbf{S}_n) - (n-2)$.

We now turn to Boolean lattices. The goal was a proof of the following conjecture.

CONJECTURE 1. *The linear extension diameter of the Boolean lattice B_n is*

$$led(B_n) = 2^{2n-2} - (n+1)2^{n-2}.$$

The conjecture is verified for $n \leq 4$ in Lemma 4.5. With the next proposition we give a construction showing that the expression given in the conjecture is a lower bound for $led(B_n)$.

PROPOSITION 4.4. $led(B_n) \geq 2^{2n-2} - (n+1)2^{n-2}$.

Proof. Let L be the reverse lexicographic order on the subsets of $[n]$; i.e., $A <_L B$ if the smallest element of the symmetric difference of A and B is in B. Clearly, L is a linear extension of B_n. Now revert the order on $1, .., n$ and let L' be the corresponding lexicographic order. L' is sometimes called the reverse antilexicographic order and can be described by$A <_{L'} B$ if the largest element of the symmetric difference is in B. Reverse lexicographic and antilexicographic order are hereditary; i.e., if $X \subset [n]$, then L restricted to the subsets of X is the reverse lexicographic order of these sets. Figure 3 displays the drawings of B_4, B_5, and B_6 obtained from this construction.

Let X be the first half of elements of L', i.e., the set of subsets of $[n]$ not containing n, and let Y be the complement of X. We count the incomparable pairs of $P_{L,L'}$ in three parts. The number of incomparable pairs (A, B) with $A \in X$ and $B \in X$ is $led(B_{n-1}) = 2^{2n-4} - n2^{n-3}$ by induction. The same is true for the pairs (A, B) with $A \in Y$ and $B \in Y$. It remains to count the incomparable pairs (A, B) with $A \in X$ and $B \in Y$; since A precedes B in L' we count pairs A, B with $n \notin A$, $n \in B$, and $B <_L A$. This number is $\binom{2^{n-1}}{2}$ since $B <_L A$ iff $A <_L (B - n)$. ∎

LEMMA 4.5. *Reverse lexicographic and reverse antilexicographic linear extensions are a diametral pair of B_n for $n \leq 4$.*

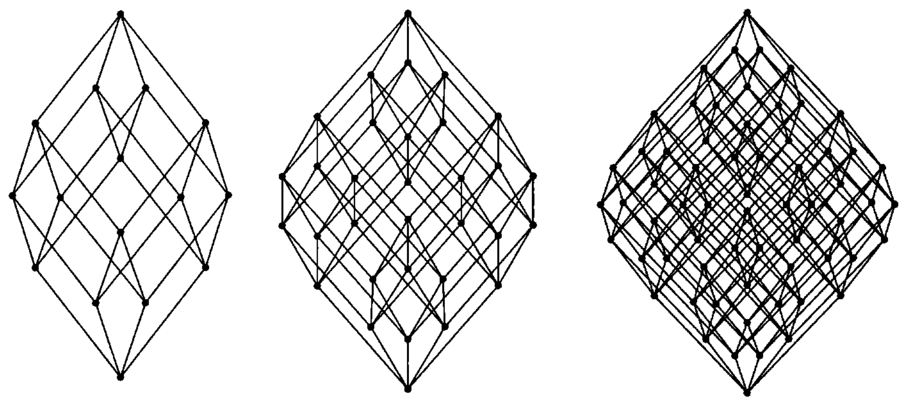


FIG. 3. *The drawing of B_4, B_5, and B_6 obtained from reverse lexicographic and reverse antilexicographic linear extensions.*

Proof. For $n \leq 2$ this is trivial. For $n = 3$ we have $8 \leq led(B_3)$ from Proposition 4.4 and $led(B_3) < inc(B_3) = 9$ from $dim(B_3) = 3$; hence the result. Now let $n = 4$. We know that at least two of the incomparabilities of the standard example S_4 contained in B_4 are comparable in the two-dimensional poset corresponding to a diametral pair. In the standard labeling of B_4 with binary vectors we may assume that these two relations are $(0100) < (1011)$ and $(0010) < (1101)$. Let $\widetilde{B_4}$ denote the poset after addition of these two relations.

Consider the following nine induced subposets of $\widetilde{B_4}$: the first is the subposet induced by $(0001), (1000), (0110), (1001), (1110), (0111)$. The other eight are denoted $Q_{i,j}$ and are obtained by inserting i at position j for $i \in \{0, 1\}$ and $j = 1, 2, 3, 4$ in each of the vectors $(001), (010), (100), (110), (101), (011)$. Each of these nine posets is a 3-crown, and it is easily checked that no two of these crowns have an incomparable min-max pair in common. It follows that in any two-dimensional extension of B_4 at least one of the three incomparable min-max pairs of each 3-crown is comparable. This gives a total of $2 + 9$ additional comparabilities in any two-dimensional extension of B_4, i.e., $led(B_4) \leq inc(B_4) - 11 = 44$. The construction of Proposition 4.4 gives a two-dimensional extension of B_4 with 44 incomparabilities which is thus optimal. □

We have not been able to generalize the proof of the previous lemma to the general case. Below we conjecture a property of diametral pairs that would imply Conjecture 1. We first state the property as Conjecture 2. Then we prove the implication in Lemma 4.6. A more detailed discussion of properties of diametral pairs will be the subject of the next section.

A *critical pair* of a poset P is an incomparable pair (x, y) such that $z < y$ implies $z < x$ and $z > x$ implies $z > y$. A critical pair (x, y) is *reverted* by linear extension L if y precedes x in L.

CONJECTURE 2. *Let L, L' be a diametral pair of a poset P. Then at least one of the two linear extensions L, L' reverts a critical pair of P.*

LEMMA 4.6. *Conjecture* 2 *implies Conjecture* 1.

Proof. Let L, L' be a diametral pair for B_n. We may assume (Conjecture 2) that L' reverts the critical pair $(\{1, .., n-1\}, \{n\})$. As in the construction we let X and Y be the sets of the first and second half of L'. Again X is the set of subsets of $[n]$ not containing n. The number of incomparable pairs (A, B) in $P_{L,L'}$ with $A \in X$ and

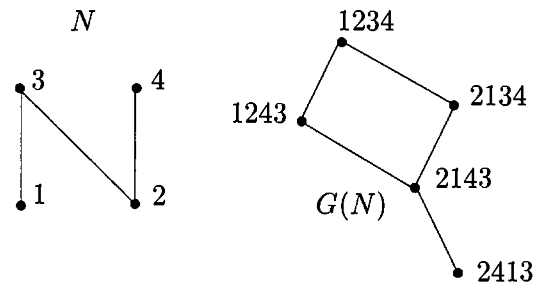


FIG. 4. *The N and its linear extension graph. The pair* (1243, 2134) *is locally extremal, and the unique extremal pair is* (1234, 2413).

$B \in X$ is at most $led(B_{n-1})$. The same holds for pairs with $A \in Y$ and $B \in Y$.

It remains to estimate the number of incomparable pairs (A, B) with $A \in X$ and $B \in Y$ that are reversed by L, i.e., pairs (A, B) with $n \notin A$, $n \in B$, and $B <_L A$. Let (A, B) be such a pair and let $mate(A, B) = (B - n, A + n)$; note that $B - n \in X$ and $A + n \in Y$. Since *mate* is an involution *mate* defines a pairing of the pairs $(A, B) \in X \times Y$. At most one of (A, B) and $mate(A, B)$ can be reversed by L; otherwise, $B <_L A <_L A + n <_L B - n <_L B$, which is a contradiction. A pair $((A, B), mate(A, B))$ that may contribute a reversal is characterized by $A, B - n$ and these are different subsets of $[n - 1]$. Therefore, the number of reversals contributed by pairs $(A, B) \in X \times Y$ is at most $\binom{|X|}{2} = \binom{2^{n-1}}{2}$. Putting things together,

$$led(B_n) \leq 2led(B_{n-1}) + \binom{2^{n-1}}{2}.$$

Induction completes the proof. □

5. Intervals in $G(P)$ and diametral pairs. For two linear extensions M, N of P let the interval $[M, N]$ in $G(P)$ consist of all linear extensions on the shortest path between M and N; put differently it is the set of linear extensions of $P_{\{M,N\}}$. We call M, N an *extremal pair* if there is no interval $[M', N']$ properly containing $[M, N]$. Note that $[M', N'] \supseteq [M, N]$ implies $dist(M', N') \geq dist(M, N)$. Hence, diametral pairs are extremal. A *locally extremal pair* is a pair M, N such that $[M, N]$ is not properly contained in $[M', N']$ with M' a neighbor of M or $M' = M$ and N' a neighbor of N or $N' = N$. Figure 4 illustrates the definitions. It is immediate that for pairs M, N of linear extensions the following implications hold:

$$\textit{diametral} \Longrightarrow \textit{extremal} \Longrightarrow \textit{locally extremal}.$$

Those diametral pairs we understand best are the minimal realizers of two-dimensional posets. Kierstead and Trotter [KT89] observed that the linear extensions of such a 2-realizer are super-greedy. The definition of greedy and super-greedy can be based on the following generic algorithm for linear extensions:

LINEAR EXTENSION
for $i = 1$ to n do
  choose $x_i \in \text{MIN}(P - \{x_1, .., x_{i-1}\})$
output $x_1, x_2, \ldots, x_n$

- For *greedy* linear extensions x_i is chosen from $\text{Min}(P-\{x_1,..,x_{i-1}\}) \cap succ(x_{i-1})$ whenever this set is nonempty.
- For *super-greedy* linear extensions x_i is chosen from $\text{Min}(P-\{x_1,..,x_{i-1}\}) \cap succ(x_j)$, where $j < i$ is maximal such that this set is nonempty.

Lemma 5.1. *Let P be a poset and L a super-greedy linear extension. Either P is a chain or L reverses a critical pair.*

Proof. We may assume that P has more than one minimal element. Let x_i be the minimal element of P that comes last in $L = x_1, \dots, x_n$. Since L is super-greedy, $P - \{x_1,..,x_i\} = succ(x_i)$ and, hence, $succ(x_{i-1}) \subseteq succ(x_i)$. Since $pred(x_i) = \emptyset \subseteq pred(x_{i-1})$, the pair (x_i, x_{i-1}) is a critical pair reversed by L. ∎

5.1. Extremal linear extensions. Call M an *extremal linear extension* if there is a linear extension N such that there is no interval $[M', N]$ properly containing $[M, N]$. Interestingly, extremal linear extension are exactly the linear extensions participating in locally extreme pairs.

Proposition 5.2. *For a linear extension M the following is equivalent:*

- *M is an extremal linear extension.*
- *There exists a linear extension N such that M, N is locally extremal.*

Proof. Let M be an extremal linear extension with witness N. We define a partial order on $G(P)$ with respect to a linear extension M as follows: $L \leq_M L'$ if the set of pairs of L' which are in reverse order relative to M contains the corresponding set for L. This is equivalent to saying that the interval $[M, L']$ contains the interval $[M, L]$. If we choose N' as a maximal element above N with respect to $\leq_M$, then M, N' is a locally extremal pair. M is extremal with respect to N' because $N' \leq_{N'} N \leq_{N'} M \leq_{N'} M'$ implies $[M, N] \subseteq [M', N]$. Since N is a witness for M's extremality, this requires $M = M'$. The other direction is obvious from the definitions. ∎

With the next proposition we characterize extremal linear extensions. Recall that a *jump* in a linear extension $L = x_1, x_2, \dots, x_n$ is a pair x_i, x_{i+1} of consecutive elements in L that are incomparable in P. If x_i, x_{i+1} are comparable in P we call the pair a *bump* of P. The *bump decomposition* of L is obtained by cutting L in each bump. This gives an ordered partition $L = \alpha_1, \alpha_2, \dots, \alpha_k$ such that each block α_i is a maximal interval of elements $x_{i_j}, .., x_{i_{j+1}-1}$ such that consecutive elements in α_i form a jump.

Example. Let P be the chevron labeled as in Figure 1. In $M = 132456$ there are three jumps and two bumps; the bumps are (24) and (56). The bump decomposition is $\alpha_1 = 132$, $\alpha_2 = 45$, $\alpha_3 = 6$.

Proposition 5.3. *A linear extension L of P is extremal iff every block α_i of the bump decomposition $\alpha_1, \alpha_2, \dots, \alpha_k$ of L induces an antichain in P.*

Proof. Let N be such that L, N is a locally extremal pair. Assume that some block α_i does not induce an antichain and let $x, y \in \alpha_i$ with $x < y$ in P. Not all the adjacent pairs of α_i can be in reverse order to N, because this would imply $y < x$ in N. Hence some adjacent pair can be switched in α_i to increase the distance to N, which is a contradiction.

In order to prove the other direction let N be the word resulting from L by reversing every block of the bump decomposition of P. If all blocks induce antichains in P, then N is a linear extension of P. Moreover, L is extremal with respect to N, since only the switch of an adjacent pair of some block yields a neighboring linear extension of L. But such a linear extension is closer to N than L is. ∎

Corollary 5.4. *Every greedy linear extension is extremal.*

Proof. If L is not extremal, then there exist x, y in some block α_i of L with x being covered by y in P. Observe that x and y cannot be adjacent in α_i. Now, L is not greedy, since y is a candidate to be chosen right after x. □

In general, however, the class of extremal linear extensions contains nongreedy linear extensions. Even both linear extensions of a locally extremal pair may be non-greedy. Take for example the 3-crown $\mathbf{C}_3^2$ on $\{0, 1, 2, 0', 1', 2'\}$ (element i' is larger than $i, i-1$) the pair $(2, 1, 0, 0', 2', 1')$, $(0, 1, 2, 1', 2', 0')$ is extremal but neither is greedy. Due to their vast amount, extremal pairs seem to be rather useless for heuristics or approximations of the linear extension diameter. In the next subsection we discuss a much stronger property.

5.2. Complementary linear extensions. Let L be a linear extension of P and specify the choice function in the algorithm Linear Extension so that in each round x_i is the last element of $\mathrm{Min}(P)$ in L; i.e., take the reverse of L as preference list for the construction of a new linear extension M. We call M the *complementary linear extension of* L and denote the complementary mapping by $*$, i.e., $* : L \to M = L^*$. The k fold iterated complementary map of L is L^{*k}.

Example. Let P be the chevron labeled as in Figure 1. If $L = 132456$, then $L^* = 315624$.

The intuition is that L^* tends to have many pairs in the reverse order of L, hence, the distance from L to L^* should be large.

Proposition 5.5. *Complementary linear extensions are super-greedy.*

Proof. Let $y_1, .., y_t$ be an initial segment of L^*. For element $x \in \mathrm{Min}(P - \{y_1, .., y_t\})$ let $i(x) = \max(i : x > y_i)$. We have to prove that y_{t+1} is an element x' with $i(x')$ maximal. Suppose that not $y_{t+1} = x'$ but that $i(x') = r < i(x) = s$. The choice of x' implies that $x <_M x'$. Consider the situation when y_s was chosen and note that at this time x' was available. Since $y_s < x$ we have $y_s <_M x'$ contradicting the choice of y_s. □

Corollary 5.6. *For linear extensions the following implications hold:*

$$\textit{complementary} \Longrightarrow \textit{super-greedy} \Longrightarrow \textit{greedy} \Longrightarrow \textit{extremal.}$$

As is the case with super-greedy linear extensions, complementary linear extensions may be constructed by an algorithm based on a stack. To construct the complementary linear extension of L begin with an empty stack S. Push the elements of $\mathrm{Min}(P)$ onto S in the order induced by L on this set. For $i = 1, .., n$ repeat: $x_i \leftarrow \mathrm{pop}(S)$ and push the new minimal elements, i.e., the elements of the set $C_i = \mathrm{Min}(P - \{x_1, .., x_i\}) - \mathrm{Min}(P - \{x_1, .., x_{i-1}\})$ onto S. The order in which elements of C_i are pushed is again the order induced by L on this set. The complementary linear extension L^* of L is $x_1, \ldots, x_n$, i.e., the elements ordered by the time of their pop. The formal proof that the stack algorithm applied to L constructs the complementary linear extension L^* is very similar to the proof of Proposition 5.5.

We illustrate the two procedures for complementary linear extensions with the following example (Table 5.1). Let P be the chevron with the labeling of Figure 1 and let $L = 132456$. In the left column of the table we have L with elements already used for L^* removed. Underlined elements are the elements of $\mathrm{Min}(P - \{x_1, .., x_{i-1}\})$, and bold are the elements of C_i, i.e., the new minimal elements. The next three columns correspond to the stack-based construction and explain themselves. Finally, there is a column with the growing L^*. We like to remark that yet another way of interpreting the construction of L^* is as a certain depth-first-search on the diagram of P with a

TABLE 5.1
Demonstrating the construction of a complementary linear extension.

L	Stack	Pop	C_i	L^*
1 **3** 2 4 5 6	13	3	$\emptyset$	3
1 2 4 5 6	1	1	$\{2,5\}$	3 1
2 4 **5** 6	25	5	$\{6\}$	3 1 5
2 4 **6**	26	6	$\emptyset$	3 1 5 6
2 4	2	2	$\{4\}$	3 1 5 6 2
4	4	4	$\emptyset$	3 1 5 6 2 4

least element 0 added. The corresponding spanning tree consists of the edges (x_i, y) for $y \in C_i$.

A *complementary pair* is a pair L, M of linear extensions with $M = L^*$ and $L = M^*$. Continuing with the example $L = 132456$, we saw $L^* = 315624$ and compute $L^2 = 125346$ and $L^{*3} = 315624$. Since $L^{*3} = L^*$, the pair L^*, L^{*2} is a complementary pair. In this case it is a diametral pair as well.

PROPOSITION 5.7. *A realizer L, L' of a two-dimensional poset is a complementary pair.*

Proof. In L' the elements of $\text{MIN}(P)$ are in the reverse of their order in L. Therefore, L' and L^* are equal in the first element x. Since $L^* = x + (L-x)^*$ and $L - x, L' - x$ is a realizer of $P - x$ induction shows $L' = L^*$. ◻

From the definition it is not obvious that every poset has a complementary pair. This, however, is an immediate consequence of the following "convergence" theorem.

THEOREM 5.8. *Let P be a poset of height h, and let L be a linear extension; then $L^{*2h-1} = L^{*2h+1}$. In other words L^{*2h-1}, L^{*2h} is a complementary pair of P.*

The proof of the theorem will be based on two lemmas.

LEMMA 5.9. *Let I be a down-set of P. The restriction of L^* to I equals the complementary linear extension of the restriction of L to the suborder induced by P on I.*

With $L|X$ denoting the restriction of L to a subset X of P, this can be written as

$$L^*|I = (L|I)^*.$$

Proof. The proof is by induction on $n = |P|$. Let x be the last minimal element of P in L and note that x is the first element of L^*. Consider $P-x$. With $M = L|(P-x)$ we have $L^* = xM^*$.

If $x \notin I$, then $M|I = L|I$ and

$$L^*|I = M^*|I = (M|I)^* = (L|I)^*,$$

with the second equality being the induction hypothesis. Otherwise, if $x \in I$, then

$$L^*|I = x\,(M^*|(I-x)) = x\,(M|(I-x))^* = x\,(L|(I-x))^* = (L|I)^*,$$

with the second equality being the induction hypothesis. ◻

LEMMA 5.10. *Let P be a poset, $A \subseteq \text{MAX}(P)$, and $Q = P - A$. If L is a linear extension of P with $L^*|Q = L^{*3}|Q$, then $L^{*3} = L^{*5}$.*

Proof. For $t \geq 1$ let $L^{*t} = x_1^t, x_2^t, \ldots, x_n^t$ and use the superscript t to denote structures involved in the stack-based construction of L^{*t}. For example, the elements of the set $C_i^t = \text{MIN}(P - \{x_1^t, .., x_i^t\}) - \text{MIN}(P - \{x_1^t, .., x_{i-1}^t\})$ are the elements pushed onto stack S^t after the pop of x_i^t.

By Lemma 5.9 $L^*|Q = L^{*3}|Q$ implies that $L^*|Q, L^{*2}|Q$ is a complementary pair for Q. If $x_i^t \notin Q$, then obviously $C_i^t = \emptyset$. Hence, for t, t' of the same parity (both odd or both even) the same sets are pushed in the same order onto the stacks S^t and $S^{t'}$. More formally, if q_i^t denotes the index of the ith element of Q in L^{*t}, then $C^t_{q_i^t} = C^{t'}_{q_i^{t'}}$ for $t = t' \bmod 2$ and $1 \le i \le |Q|$. Using the simplified notation $\mathcal{C}_i^t = C^t_{q_i^t}$ (with calligraphic $\mathcal{C}$) we restate this fact.

Fact. $\mathcal{C}_i^t = \mathcal{C}_i^{t'}$ for $t = t' \bmod 2$ and $1 \le i \le |Q|$.

The linear extension L^{*t} is completely determined by the evolution of the stack S^t. From $\mathcal{C}_i^t = \mathcal{C}_i^{t'}$ we could conclude that L^{*t} depends only on the parity of t if the order in which the elements of $\mathcal{C}_i^t$ are pushed onto S^t remained unchanged or, equivalently, if the order of the elements of $\mathcal{C}_i^t$ in L^{*t} remained unchanged. This will be proved for $t \ge 3$.

Let $D_{ij} = \mathcal{C}_i^1 \cap \mathcal{C}_j^2 = \mathcal{C}_i^o \cap \mathcal{C}_j^e$ for o odd and e even and note that there is an order α_{ij} of the elements of D_{ij} such that in the sequence L^{*t} the order of these elements alternates between α_{ij} for t odd and the reverse of α_{ij} for t even.

Claim. Let $j < k$ and $y \in D_{ij}$, $x \in D_{ik}$. For $t \ge 3$, t odd, x precedes y in L^{*t}.

Proof of claim. Assume the existence of $o \ge 3$ odd such that y precedes x in L^{*o}; we shorten notation, writing $y <_o x$ for this fact. Since $x, y \in \mathcal{C}_i^o$ we conclude that $x <_{o-1} y$. Let $e = o-1$ and recall $j < k$ and $y \in \mathcal{C}_j^e$ and $x \in \mathcal{C}_k^e$. Hence, y was pushed onto stack S^e earlier than x, and since $x <_e y$, element y was still buried in S^e when x was pushed. Inspection shows that there was a $z \in \mathcal{C}_j^e$ with $z < x$ and z was pushed after y onto S^e. It follows that the order of x, y, z in L^{*e-1} is $y <_{e-1} z <_{e-1} x$.

From $x, y \in \mathcal{C}_i^{e-1} = \mathcal{C}_i^o$ and $y <_{e-1} x$, we obtain that x was pushed before y onto S^{e-1}. Since $z < x$, element z was pushed onto S^{e-1} before x and y.

To obtain $y <_{e-1} z <_{e-1} x$ the stack S^{e-1} would thus get the elements pushed in order z, x, y and pop them off in order y, z, x. This, however, corresponds to a 3-element permutation that cannot be realized with a stack. This contradiction concludes the proof of the claim.

It follows that for $t \ge 3$, t odd the order of the elements of $\mathcal{C}_i^o$ in L^{*t} is $\alpha_{i,n-1} <_t \alpha_{i,n-2} <_t \cdots <_t \alpha_{i,1}$. This completely determines the evolution of the stack; hence, $L^{*3} = L^{*5} = L^{*7} \ldots$. □

Proof of Theorem 5.8. Let $A_1, A_2, \ldots, A_h$ be the canonical antichain partition of P with *height*$(P) = h$; i.e., $A_{i+1} = \mathrm{Min}(P - A_1 - \cdots - A_i)$ and $\bigcup_1^h A_i = P$. Let $A_{\le k} = A_1 \cup A_2 \cup \cdots \cup A_k$ and note that $A_{\le k}$ is a down-set.

Claim. $L^{*2k-1}|A_{\le k} = L^{*2k+1}|A_{\le k}$ for $k = 1, \ldots, h$.

Proof of claim. By Lemma 5.9 it suffices to prove $(L|A_{\le k})^{*2k-1} = (L|A_{\le k})^{*2k+1}$.

For $k = 1$ this is trivially true. Since $A_k \subseteq \mathrm{Max}(A_{\le k})$ we can use Lemma 5.10 with $L = L^{*2k-4}|A_{\le k}$ for the induction step, proving the claim.

Since $A_{\le h} = P$ this implies the theorem. □

Proposition 5.11. *If M, N is a complementary pair, then the interval* $[M, N]$ *is locally extreme in* $G(P)$.

Proof. Assume that there is neighbor N' of N such that $[M, N] \subset [M, N']$. Let (x, y) be the unique pair with $x <_N y$ and $y <_{N'} x$. Since $N = M^*$ and both x and y were minimal elements when x was chosen we find that $y <_M x$. This implies that N' is on a shortest path from M to N, which is a contradiction to $[M, N] \subset [M, N']$. Similar arguments disprove the other cases. □

A diametral pair need not be a complementary pair. An example is given in Figure 5.

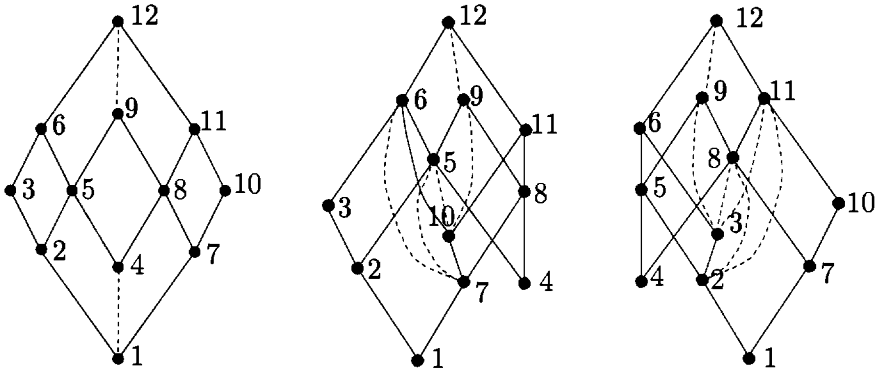


FIG. 5. *Left: P and its unique minimum two-dimensional extension. Middle and right: The two complementary two-dimensional extensions of P.*

Acknowledgment. We thank Michael Naatz for careful proofreading and suggesting some improvements.

REFERENCES

[BS92] G. Brightwell and E. Scheinerman, *Fractional dimension of partial orders*, Order, 9 (1992), pp. 139–158.

[BW91] A. Björner and M. Wachs, *Permutation statistics and linear extensions of posets*, J. Combin Theory Ser. A, 58 (1991), pp. 85–114.

[DPW85] B. Dreesen, W. Poguntke, and P. Winkler, *Comparability invariance of the fixed point property*, Order, 2 (1985), pp. 269–274.

[Gal67] T. Gallai, *Transitiv orientierbare Graphen*, Acta. Math. Hungar., 18 (1967), pp. 25–66.

[KT89] H. Kierstead and W.T. Trotter, *Super-greedy linear extensions of ordered sets*, Ann. New York Acad. Sci., 555 (1989), pp. 262–271.

[Reu96a] K. Reuter, *The Comparability Graph and the Graph of Linear Extensions of a Poset*, Preprint, Universität Hamburg, 1996.

[Reu96b] K. Reuter, *Linear Extensions of a Poset as Abstract Convex Sets*, Preprint, Universität Hamburg, 1996.

[Tro92] W.T. Trotter, *Combinatorics and Partially Ordered Sets: Dimension Theory*, The Johns Hopkins Press, Baltimore, MD 1992.

[TW96] W.T. Trotter and P. Winkler, *Ramsey theory and sequences of random variables*, Combin. Probab. Comput., 7 (1998), pp. 221–238.

SIAM J. DISCRETE MATH.
Vol. 12, No. 3, pp. 374–384

RANKINGS OF DIRECTED GRAPHS*

JAN KRATOCHVÍL† AND ZSOLT TUZA‡

Abstract. A ranking of a graph is a coloring of the vertex set with positive integers in such a way that on every path connecting two vertices of the same color there is a vertex of larger color. We consider the directed variant of this problem, where the above condition is imposed only on those paths in which all edges are oriented consecutively. We show that the ranking number of an orientation of a tree is bounded by that of its longest directed path plus one, and that it can be computed in polynomial time. Unlike the undirected case, however, deciding whether the ranking number of a directed (and even of an acyclic directed) graph is bounded by a constant is NP-complete. In fact, the 3-ranking of planar bipartite acyclic digraphs is already hard.

Key words. graph, directed graph, oriented graph, ranking, algorithm, NP-completeness

AMS subject classifications. 05C15, 05C20, 05C05, 05C85

PII. S0895480197330242

1. Introduction. Given an undirected graph G, its *ranking number* $\chi_r(G)$ is the minimum integer k for which there exists a *(vertex) k-ranking*; that is, a mapping $f : V(G) \to \{1, 2, \ldots, k\}$ such that every path connecting two vertices u, v of the same rank $f(u) = f(v)$ contains a vertex w with higher rank, $f(w) > f(u)$. In particular, adjacent vertices have to be assigned different ranks; therefore $\chi_r(G) \geq \chi(G)$ for every graph. As a variant of graph coloring problems, vertex and edge rankings find applications in scheduling, very large scale integration design, and notably in parallel matrix-factorization algorithms (see, e.g., [1] for references).

It is well known [1] that for the path P_ℓ of length $\ell - 1$ on ℓ vertices,

$$\chi_r(P_\ell) = \lfloor \log \ell \rfloor + 1$$

holds, and the longest k-rankable path $P_{2^k-1} = x_1 x_2 \ldots x_{2^k-1}$ admits the *unique* optimal ranking f with

$$f(x_i) = \max\{j : i \equiv 0 \bmod 2^j\} + 1$$

for all $1 \leq i < 2^k$. (Throughout, log means *logarithm of base* 2.) For the proof of uniqueness, note that the highest rank k appears exactly once; otherwise the subpath connecting two vertices of rank k would not contain a vertex of higher rank. Applying induction, it also follows that rank k is assigned to the middle vertex and the subpaths on its two sides are uniquely $(k-1)$-ranked.

*Received by the editors November 19, 1997; accepted for publication (in revised form) March 3, 1999; published electronically September 7, 1999. A preliminary version of this article appeared as *Rankings of directed graphs*, in Graph-Theoretic Concepts in Computer Science, Proceedings of the 24th International Workshop WG '98, Smolenice Castle, Slovak Republic, June 18–20, 1998, Lecture Notes in Comput. Sci. 1517, Springer-Verlag, Berlin, 1998, pp. 114–123.

http://www.siam.org/journals/sidma/12-3/33024.html

†Department of Applied Mathematics, Charles University, Malostranské nám. 25, 118 00 Praha 1, Czech Republic (honza@kam.ms.mff.cuni.cz). The research of this author was supported in part by Czech research grants GAUK 194/1996, GAUK 158/1999, and GAČR 201/1996/0194.

‡Computer and Automation Institute, Hungarian Academy of Sciences, H–1111 Budapest, Kende u. 13–17, Hungary (tuza@sztaki.hu). The research of this author was supported in part by the Hungarian Scientific Research Fund, grants OTKA T–016416 and T–026575, and by Czech grant GAČR 201/1999/0242.

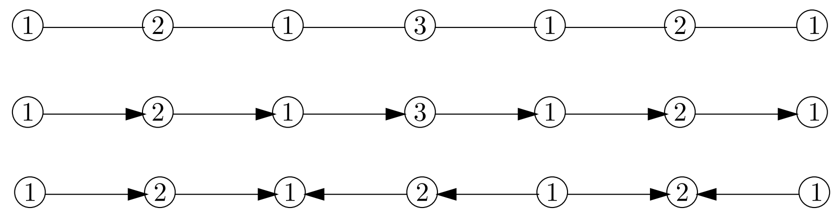


FIG. 1.1. *Undirected and directed rankings of paths.*

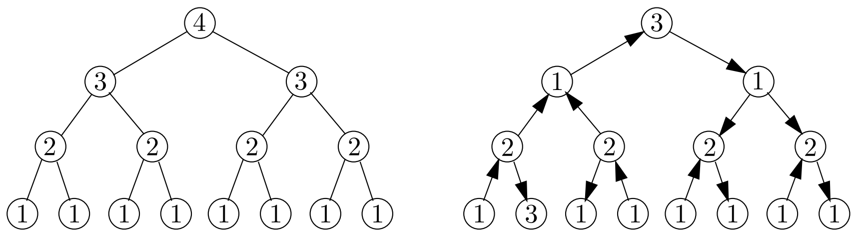


FIG. 1.2. *Undirected and directed ranking of trees.*

Recently, oriented chromatic number was introduced and intensively studied as a variant of chromatic number for directed graphs [5, 9, 11, 10, 13]. In the same spirit, it is natural to study the effect of orientations on rankings as well. This paper is the first approach to the ranking of *directed* graphs. The ranking number of a digraph G is naturally defined as the minimum k such that there exists a mapping $f : V(G) \to \{1, 2, \ldots, k\}$ with the property that every *directed path* (i.e., path in which all edges are oriented consecutively) connecting two vertices u, v of the same rank $f(u) = f(v)$ contains a vertex w with higher rank, $f(w) > f(u)$. We denote the ranking number of a directed graph G again by $\chi_r(G)$.

Obviously, the ranking number of a directed path equals that of the undirected path of the same length. Rankings of undirected and oriented paths are shown in Figure 1.1.

Directed and undirected rankings, however, have a strikingly different behavior already on trees. For instance, an undirected tree containing no path longer than t can have a ranking number as large as $\lceil t/2 \rceil + 1$. (One can show that the level-wise ranking of a complete binary tree is optimal, which shows that the tree of height $t/2$ has longest path of length t and ranking number $t/2 + 1$.) This is far from being true in the directed case. We shall prove that the ranking number of any orientation of a tree can exceed that of its longest directed path by at most 1 (Corollary 2.3), hence it grows just with $\log t$. Examples of ranking an undirected complete binary tree and an orientation of it are given in Figure 1.2.

We also consider rankings from the computational complexity point of view. The problem RANKING takes as input a graph G and a positive integer k and asks whether $\chi_r(G) \le k$. It is known that RANKING on undirected graphs is NP-complete in general but solvable in polynomial time for every fixed k; see [1] for results and further references. For the analogous problem of DIRECTED RANKING, however, we prove in Theorem 4.1 that it is NP-complete even if the input is restricted to fixed $k = 3$ and to *acyclic* orientations of *planar bipartite* graphs. On the other hand, the 2-rankable directed graphs can be characterized in several different ways, as shown in section 5. We also prove that the ranking number of oriented *trees* can be determined in polynomial time (section 3).

The study of directed ranking is an appealing task in itself, leading to interesting theoretical problems. It is worth mentioning, however, that in the context of the *original motivation* for introducing the ranking number of graphs in [4], the directed version defined in our present work may appear at least as natural as the undirected one. Formally, the most general setting would be the study of rankings in partially oriented graphs. This can be modeled, however, by introducing two oppositely oriented arcs for each undirected edge of the input graph, and hence it turns out to be equivalent to directed ranking in general digraphs (not just in oriented graphs). Our results in section 4 provide relevant information on this more general problem as well.

2. Upper bound for trees. In this section we prove general bounds on the ranking number of oriented trees and also on that of orientations of a path of given length. We begin with some definitions.

Notation. We write $p(\ell) := \lfloor \log \ell \rfloor + 1 = \chi_r(P_\ell)$ for the ranking number of the (directed or undirected) path with ℓ vertices (i.e., $p(\ell) = k$ if and only if $2^{k-1} \leq \ell \leq 2^k - 1$). Moreover, we define $r_t(\ell)$ and $r_p(\ell)$ as the maximum ranking number of orientations of trees and that of orientations of undirected paths, respectively, under the condition that *no directed subpath has more than* ℓ *vertices*.

Clearly, $r_t(\ell) \geq r_p(\ell) \geq p(\ell)$ holds for all $\ell \geq 1$. Our results will show that these three functions are very close to each other, in the entire range of ℓ.

THEOREM 2.1. *For every* $k \geq 2$ *and* ℓ *such that* $2^{k-2} + 1 \leq \ell \leq 2^{k-1}$,

$$r_t(\ell) = k\,.$$

Proof. We first show that $\chi_r(T) \leq k$ holds, provided that every directed subpath of a given oriented tree T has at most 2^{k-1} vertices. Consider an infinite directed path with vertices x_i and edges $x_i x_{i+1}$, $i \in \mathbf{Z}$. Define a mapping $\phi : \{x_i : i \in \mathbf{Z}\} \to \{1, 2, \ldots, k\}$ by

$$\phi(x_i) = \begin{cases} k & \text{if } i \equiv 0 \bmod 2^{k-1}\,, \\ \max\{j : i \equiv 0 \bmod 2^{j-1}\} & \text{if } i \not\equiv 0 \bmod 2^{k-1}\,. \end{cases}$$

Obviously, any segment of at most 2^{k-1} vertices is ranked in a feasible way by ϕ.

Now we consider an oriented tree T containing no directed subpath with more than 2^{k-1} vertices. We view such a tree as a Hasse diagram of a partially ordered set and, as such, partition its vertex set into levels: we choose an arbitrary vertex and call its level $L(0)$, and then we recursively sort the other vertices—a vertex u is placed into level $L(i+1)$ if there is a vertex v already in level $L(i)$ such that $uv \in E(T)$, and a vertex w is placed into level $L(i-1)$ if there is a vertex v already in level $L(i)$ such that $vw \in E(T)$. A mapping f defined by $f(u) = \phi(x_i)$ for $u \in L(i)$ is then a feasible k-ranking of T. (The above procedure partitions T into levels correctly, since T is a tree.)

We next turn to the lower bound for $r_t(\ell)$, namely, $r_t(2^{k-1}+1) > k$. For $i = 2^{k-1}$, $2^{k-1}-1, \ldots, 3, 2, 1, 0$, we construct a series of trees $T_k(i)$ recursively (in this decreasing order of i), with the following properties:

(1) Every directed subpath of $T_k(i)$ has at most $2^{k-1} + 1$ vertices;
(2) $T_k(i)$ contains a nonextendable directed path P of length $2^{k-1}-1$ with vertices $x_1, x_2, \ldots, x_{2^{k-1}}$ and arcs $x_h x_{h+1}$, $1 \leq h < 2^{k-1}$;
(3) for every $j \leq i$, every directed path of $T_k(i)$ passing through x_j has at most 2^{k-1} vertices; and

(4) for every feasible k-ranking f of $T_k(i)$ and for every $j > i$, $f(x_j) \neq k$.

The first step of the construction is for $i = 2^{k-1}$, and for $T_k(2^{k-1})$ we simply take the path $P = x_1x_2\dots x_{2^{k-1}}$. In the recursive step, we take a copy T' of $T_k(i+1)$ with vertex set disjoint from the vertex set of $T_k(i+1)$ and add the arc $x'_{i+1}x_{i+1}$ to the disjoint union of T' and $T_k(i+1)$. (We assume that the copy of P is denoted by $P' = x'_1x'_2\dots x'_{2^{k-1}}$ in T'.) This will be our $T_k(i)$, and $P = x_1x_2\dots x_{2^{k-1}}$ will keep playing the role of the path P for property (2).

The properties (1)–(3) for $T_k(i)$ clearly follow by induction. To prove (4), we revoke the result known from undirected ranking: the longest $(k-1)$-rankable path has $2^{k-1}-1$ vertices. Hence, in any feasible k-ranking f_{i+1} of $T_k(i+1)$, at least one of the vertices of P is ranked k. If f_i is a k-ranking of $T_k(i)$, by the induction hypothesis none of the vertices x'_j, $j > i+1$, is ranked k, and hence at least one of the vertices x'_j, $1 \leq j \leq i+1$, is ranked k. On the other hand, the directed path $x'_1\dots x'_{i+1}x_{i+1}\dots x_{2^{k-1}}$ contains at most one vertex ranked k, and thus property (4) follows for $T_k(i)$.

The tree $T = T_k(0)$ has no directed path with more than $2^{k-1}+1$ vertices and it is not k-rankable. Indeed, if f_0 were a feasible k-ranking, then property (4) would imply that no vertex of P is ranked k, contradicting the fact that the path with 2^{k-1} vertices is not $(k-1)$-rankable. Thus, $r_t(2^{k-1}+1) \geq k+1$. ∎

Next, we show that the ranking number of oriented trees of maximum degree 2 (i.e., orientations of undirected paths) usually equals the ranking number of their longest paths.

THEOREM 2.2. *For every $k \geq 3$ and every ℓ such that $2^{k-1}-1 \leq \ell \leq 2^k-2$,*

$$r_p(\ell) = k\,.$$

Proof. We first prove the upper bound, i.e., $r_p(2^k-2) \leq k$. It is easy to see that every (directed or undirected) path with at most 2^k-2 vertices has a feasible k-ranking such that the first vertex is ranked 1 and the last vertex is ranked 2. Thus, if T is an orientation of a path consisting of several segments of length at most 2^k-3 (a segment is a maximal directed subpath), we can k-rank each segment separately so that the sources are ranked 1 and the sinks are ranked 2.

On the other hand, to show the lower bound, we take two vertex-disjoint paths of length 2^k-2 each and orient an arc from the first vertex of one of them to the last vertex of the other one. The resulting graph has no feasible k-ranking, because in every k-ranking of a directed path of length 2^k-2, both endvertices are ranked 1; thus the added arc would connect two vertices ranked 1, which is a contradiction. Therefore $r_p(2^k-1) \geq k+1$. ∎

Reformulating the results proven above, and relating the ranking number of an oriented tree to the ranking number of its longest paths, we obtain the following corollary.

COROLLARY 2.3. *The ranking number of an oriented tree is always less than or equal to the ranking number of its longest directed paths plus* 1. *This bound is best possible, as*

$$r_t(\ell) = \begin{cases} p(\ell) & \text{if } \ell = 2^k \text{ for some } k,\\ p(\ell)+1 & \text{otherwise.} \end{cases}$$

Similarly, for orientations of undirected paths, we have

$$r_p(\ell) = \begin{cases} p(\ell)+1 & \text{if } \ell = 2^k-1 \text{ for some } k,\\ p(\ell) & \text{otherwise.} \end{cases}$$

3. Algorithm for trees. In this section we prove that the ranking number of an oriented tree can be determined by a polynomial-time algorithm. Assuming that a natural number k and a tree T with n vertices are given, our goal here is to decide by an efficient algorithm if $\chi_r(T) \leq k$. At the end of the section we shall indicate how the methods can be extended for graphs of bounded treewidth, too.

We shall use the following notation. We pick a vertex, say r, and regard T as rooted in r. (This does not mean, however, that all edges of T are necessarily oriented towards r.) For a vertex u of T, denote by T_u the subtree rooted in u and induced by those vertices from which the path (in the underlying undirected graph of T) to r passes through u. If $u \neq r$, then u^+ denotes the first vertex on the path from u to the root r. The vertices adjacent to u other than u^+ are called the children of u. We denote by $Ch_\rightarrow(u) = \{x : xu \in E(T) \wedge x \neq u^+\}$ the set of children of u for which the edges are oriented toward u, and by $Ch_\leftarrow(u) = \{x : ux \in E(T) \wedge x \neq u^+\}$ the set of those children for which the edges are oriented from u.

Given a coloring f of the vertices of T with colors $\{1, 2, \ldots, k\}$, we say that a color i is *up-visible from u* if T_u contains a directed path $P = u_1 \ldots u_j$, $u_j = u$, such that $f(u_1) = i$ and no vertex u_ℓ, $\ell = 2, \ldots, j$, is colored with a color higher than i. Similarly, a color i is *down-visible from u* if T_u contains a directed path $P = u_1 \ldots u_j$, $u_1 = u$, such that $f(u_j) = i$ and no vertex u_ℓ, $\ell = 1, \ldots, j-1$, is colored with a color higher than i. The following obvious proposition is a key observation for the forthcoming algorithm.

PROPOSITION 3.1. *For all $v \in Ch_\rightarrow(u) \cup Ch_\leftarrow(u)$, let f_v be rankings of the T_v's, the subtrees rooted in the children of u. Then for any color i, the coloring f defined by*

$$f(x) = \begin{cases} i & \text{for } x = u, \\ f^v(x) & \text{if } x \in T_v \text{ for some child } v \text{ of } u \end{cases}$$

is a proper ranking of T_u if and only if

(1) *i is not down-visible in f_v from v for any $v \in Ch_\leftarrow(u)$,*
(2) *i is not up-visible in f_v from v for any $v \in Ch_\rightarrow(u)$,*
(3) *if j is down-visible in f_v from v for some $v \in Ch_\leftarrow(u)$ and up-visible in $f_{v'}$ from v' for some $v' \in Ch_\rightarrow(u)$, then $j < i$.*

The algorithm described below scans recursively the vertices of T from the leaves to the root and computes set systems $Vis_\rightarrow(u), Vis_\leftarrow(u)$ for every $u \in V(T)$. Each of these systems is a family of subsets of $\{1, 2, \ldots, k\}$, storing essential information concerning the feasible rankings of the subtree rooted in u. Namely, $S \in Vis_\rightarrow(u)$ if and only if T_u admits a ranking such that S is the set of colors up-visible from u, and $S \in Vis_\leftarrow(u)$ if and only if T_u admits a ranking such that S is the set of colors down-visible from u.

It follows directly from Proposition 3.1 how the values of $Vis_\leftarrow(u)$ and $Vis_\rightarrow(u)$ can be computed from the values on the children of u.

PROPOSITION 3.2. *For any vertex u of T, the sets $Vis_\leftarrow(u)$ and $Vis_\rightarrow(u)$ are nonempty if and only if there exist a color i and sets $S^v_\leftarrow \in Vis_\leftarrow(v)$ (for all $v \in Ch_\leftarrow(u)$) and $S^v_\rightarrow \in Vis_\rightarrow(v)$ (for all $v \in Ch_\rightarrow(u)$) such that*

(1) $i \notin \bigcup_{v \in Ch_\leftarrow(u)} S^v_\leftarrow$,
(2) $i \notin \bigcup_{v \in Ch_\rightarrow(u)} S^v_\rightarrow$, *and*
(3) $\max\left(\bigcup_{v \in Ch_\leftarrow(u)} S^v_\leftarrow \cap \bigcup_{v \in Ch_\rightarrow(u)} S^v_\rightarrow\right) < i$.

In this case,

$$\left(\bigcup_{v \in Ch_{\leftarrow}(u)} S^v_{\leftarrow} \cap \{i+1, \ldots, k\} \right) \cup \{i\} \in Vis_{\leftarrow}(u)$$

and

$$\left(\bigcup_{v \in Ch_{\rightarrow}(u)} S^v_{\rightarrow} \cap \{i+1, \ldots, k\} \right) \cup \{i\} \in Vis_{\rightarrow}(u)\,.$$

By the above propositions, the correctness of the next algorithm follows.

ALGORITHM TREE.

Input: An oriented tree T and a number k.
0. Choose a vertex r of T and consider it the root of T.
1. For every leaf u of T and for every $i = 1, 2, \ldots, k$ set
$Vis_{\leftarrow}(u) := \{\{1\}, \{2\}, \ldots, \{k\}\}$ and
$Vis_{\rightarrow}(u) := \{\{1\}, \{2\}, \ldots, \{k\}\}$.
2. For every inner vertex u of T, in postorder do
2.1. $Vis_{\leftarrow}(u) := \emptyset$ and $Vis_{\rightarrow}(u) := \emptyset$;
2.2. $Down(u) := \{\bigcup_{v \in Ch_{\leftarrow}(u)} S_v \ : \ S_v \in Vis_{\leftarrow}(v)\}$;
2.3. $Up(u) := \{\bigcup_{v \in Ch_{\rightarrow}(u)} S_v \ : \ S_v \in Vis_{\rightarrow}(v)\}$;
2.4. for every $A \in Down(u)$ and $B \in Up(u)$ do
for every $i > \max Down(u) \cap Up(u)$ do
if $i \notin A \cup B$ then $Vis_{\leftarrow}(u) := Vis_{\leftarrow}(u) \cup \{(A \cap \{i, i+1, \ldots, k\}) \cup \{i\}\}$;
if $i \notin A \cup B$ then $Vis_{\rightarrow}(u) := Vis_{\rightarrow}(u) \cup \{(B \cap \{i, i+1, \ldots, k\}) \cup \{i\}\}$;
3. If $Vis_{\leftarrow}(r) = Vis_{\rightarrow}(r) = \emptyset$
then **output** $\chi_r(T) > k$
else **output** $\chi_r(T) \leq k$.

(In step 2.3 we use $\max \emptyset = 0$.) To show that this algorithm can be implemented in polynomial time it remains to show how steps 2.2 and 2.3 can be implemented. Note that the role of these steps is to compute the sets of all unions of elements of $Vis_{\leftarrow}(v)$ for $v \in Ch_{\leftarrow}(u)$ (or of $Vis_{\rightarrow}(v)$ for $v \in Ch_{\rightarrow}(u)$, respectively), taking one element from each Vis-set in all possible ways. This may in principle cover an exponential number of possibilities (exponential in the degree of u). This can be avoided by using dynamic programming as in the following subroutines.

FUNCTION $Up(u)$.

Let $Ch_{\rightarrow}(u) = \{u_1, u_2, \ldots, u_s\}$.
$Up := \{\emptyset\}$;
for $j := 1$ **to** s **do** $Up := \{A \cup B : A \in Up, B \in Vis_{\rightarrow}(u_j)\}$.

FUNCTION $Down(u)$.

Let $Ch_{\leftarrow}(u) = \{u_1, u_2, \ldots, u_t\}$.
$Down := \{\emptyset\}$;
for $j := 1$ **to** t **do** $Down := \{A \cup B : A \in Down, B \in Vis_{\leftarrow}(u_j)\}$.

PROPOSITION 3.3. *The running time of the algorithm* TREE(k) *is at most* $cnk^2\, 2^{2k}$*, for some absolute constant* c *independent of* k.

Proof. The function Up (which is a dynamic programming version for computing the set of all unions of type $\bigcup_{j=1}^{s} A_j$ for $A_j \in Vis_{\rightarrow}(u_j)$) needs at most 2^{2k} set unions in each of the s steps. Hence, on a vertex with s ingoing children, Up runs in $O(sk\, 2^{2k})$ time. The analogous property holds for $Down$ as well. Throughout the entire tree

T, there are as many children of processed vertices as the number of edges of T, and therefore *Up* and *Down* will consume in total at most $O(nk\,2^{2k})$ steps.

Step 2.4 requires at most $O(k^2\,2^{2k})$ time for u fixed and, being performed for every vertex, it takes at most $O(nk^2\,2^{2k})$. □

In conclusion, we obtain the following theorem.

THEOREM 3.4. *For any oriented tree T on n vertices, the directed ranking number of T can be determined in time $O(n\,\ell^2\log^3\ell)$, where $\ell \geq 2$ is the length of a longest directed path in T.*

Proof. Assume $n \geq 2$. We know from Theorem 2.2 that $1 \leq \chi_r(T) - 1 \leq \log\ell$. Therefore, it suffices to run the algorithm TREE(k) for at most $\log\ell$ values of $k \leq \log\ell + 1$, and for each of them, TREE(k) takes at most $O(n\cdot\log^2\ell\cdot 2^{2\log\ell}) = O(n\cdot\ell^2\cdot\log^2\ell)$ time. □

It was not our aim to present an optimal algorithm. Note that various further tricks can be implemented to achieve slightly better running time. Note also that binary search can be used in order to determine the ranking number of an oriented tree in time $O(n\,\ell^2\log^2\ell\,\log\log\ell)$ and that our algorithm also can be used for partially oriented trees. The above results can be extended to the following, more general theorem whose proof is only sketched. (For the definition of treewidth, see, e.g., [1].)

THEOREM 3.5. *For every fixed natural number t, the directed ranking number can be determined in polynomial time for any digraph whose underlying undirected graph has treewidth at most t.*

Proof (sketch). Recall first that the ranking number of a digraph does not exceed the ranking number of its underlying undirected graph and, second, that this is $\leq c\log n$ for graphs of bounded treewidth, say, $w(G) \leq t$ [1]. Note that each vertex u of the decomposition tree corresponds to a cut set X_u in G of size $\leq t+1$. The decomposition tree is considered rooted, and for the node u, G_u denotes the subgraph of G induced by $\bigcup X_v$, where this union is taken over all nodes v such that u lies on the path from the root to v.

The algorithm would scan recursively a tree-decomposition of G, storing for every vertex u of the tree of the tree-decomposition the characteristics of all possible rankings of G_u. For particular u, the characteristics of a ranking f of G_u consist of

(1) the restriction of the ranking f to X_u;

(2) for every vertex $v \in X_u$, the sets of up- and down-visible colors on paths leading to and from G_u;

(3) for every color i the digraph on X_u such that xy is an arc if and only if there exists a directed path in G_u from x to y whose vertices are all ranked $\leq i$.

It is only a matter of a technical case analysis to show that the information on characteristics can be propagated through the decomposition tree in polynomial time. And obviously, $\chi_r(G) \leq k$ if and only if the root of G allows a nonempty set of characteristics. The amount of stored information is at most $k^{t+1}(2^k)^{2t+2}k^{t^2} \leq O(n^{2(t+1)c}\log^{t^2+t+1}n)$. □

4. Ranking number of bipartite acyclic digraphs. Here we consider the algorithmic problem on directed acyclic graphs (DAGs).

THEOREM 4.1. *The problem* DIRECTED RANKING *is NP-complete on DAGs with planar bipartite underlying graphs, even for fixed $k = 3$.*

Proof. We show a reduction from a variant of the PRECOLORING EXTENSION problem of undirected graphs. It is known [6] that the following problem is NP-complete:

Given a planar bipartite graph with some of its vertices properly colored with three colors, does G admit a proper 3-coloring that extends the precoloring?

Given such a bipartite graph $G = (A \cup B, E)$, observe that we may assume without loss of generality that all the precolored vertices belong to A. Indeed, for each precolored vertex $v \in B$, we create two new precolored vertices of degree 1, adjacent to v, and precolor them with the two colors different from the one prescribed for v; then v can be made precolorless, as its precolored pendant neighbors force it to get the originally prescribed color.

Given a planar bipartite graph $G = (A \cup B, E)$ with precolored vertex set $Z \subseteq A$ and precoloring $\phi : Z \to \{1, 2, 3\}$, we construct a directed graph D with vertex set

$$V(D) = A \cup B \cup \{z_i^j : z \in Z,\ 1 \leq i \leq 7,\ 1 \leq j \leq 2\}$$

and arc set

$$E(D) = \bigcup_{\substack{u \in A,\, v \in B \\ uv \in E}} \{uv\} \ \cup \bigcup_{\substack{z \in Z \\ 1 \leq i \leq 6 \\ 1 \leq j \leq 2}} \{z_i^j z_{i+1}^j\} \ \cup \bigcup_{z \in Z} \{z z_{i_1(z)}^1, z z_{i_2(z)}^2\},$$

where

$$i_1(z) = \begin{cases} 6 & \text{if } \phi(z) = 1, \\ 7 & \text{if } \phi(z) = 2 \vee 3, \end{cases} \qquad i_2(z) = \begin{cases} 4 & \text{if } \phi(z) = 1 \vee 2, \\ 6 & \text{if } \phi(z) = 3. \end{cases}$$

Obviously, D is acyclic, and it also remains planar and bipartite because so is G. We claim that D is 3-rankable if and only if G admits a precoloring extension with three colors.

Suppose first that D is 3-rankable, and let $f : V(D) \to \{1, 2, 3\}$ be a feasible ranking. Since the paths $P_{z,j} = z_1^j z_2^j \dots z_7^j$ ($z \in Z$, $j = 1, 2$) are uniquely 3-rankable induced subgraphs of D, we must have $f(z_1^j) = f(z_3^j) = f(z_5^j) = f(z_7^j) = 1$, $f(z_2^j) = f(z_6^j) = 2$, and $f(z_4^j) = 3$. In this way, each $P_{z,j}$ excludes one well-defined color from its neighbor in A, and the total effect is that precisely the two colors distinct from $\phi(z)$ get excluded at each $z \in Z$. It follows that $f(z) = \phi(z)$ holds, and therefore f is a proper 3-coloring of G extending the precoloring ϕ.

On the other hand, any proper precoloring extension of ϕ together with the color sequence 1213121 on each $P_{z,j}$ gives a feasible 3-ranking. ∎

COROLLARY 4.2. *For every fixed ranking number $k \geq 3$, the problem* DIRECTED RANKING *is NP-complete on planar acyclic digraphs.*

Proof. Take a DAG, G, whose $\chi_r(G) \leq 3$ is questioned. For every vertex $x \in V(G)$, add a directed path $P_x = x_1 x_2 \dots x_{2^k - 1}$ and join x to vertices x_i such that $i \equiv 0 \bmod 8$ by arcs directed from x towards P_x. In any k-ranking of this planar DAG G', the vertices of each P_x are uniquely ranked, and the neighbors of x receive all colors $4, 5, \dots, k$. Therefore, x must be colored 1, 2, or 3. Hence, G' is k-rankable if and only if G is 3-rankable. ∎

5. Directed 2-rankable graphs. Here we investigate directed rankings with $k = 2$ colors. For the structural characterization of 2-rankable digraphs the following concept will be convenient to introduce. By an *alternating walk of length ℓ* we mean a sequence $P = x_0 x_1 \dots x_\ell$ of (not necessarily distinct) vertices such that its orientation is $x_0 \to x_1 \leftarrow x_2 \to x_3 \leftarrow \cdots$, i.e., $x_{2i} x_{2i+1} \in E$ for all $0 \leq i < \ell/2$ and $x_{2i} x_{2i-1} \in E$ for all $1 \leq i \leq \ell/2$. An alternating walk is an *alternating path* if its vertices are mutually distinct. Moreover, we say that a vertex v is *starting*, *central*, or *ending*, if there is a *directed path* $P_3 = x_1x_2x_3$ with $x_1 = v$, $x_2 = v$, or $x_3 = v$, respectively. In the present context, alternating paths and cycles of *odd* lengths will be crucial.

THEOREM 5.1. *For every digraph $G = (V, E)$, the following conditions are equivalent:*

(1) *G is 2-rankable.*
(2) *G contains no alternating path of odd length from a starting vertex to an ending vertex.*
(3) *G contains no alternating walk of odd length with both endpoints being central vertices.*
(4) *G admits a proper 2-coloring in which the set of central vertices is monochromatic.*

Proof.

(1) ⇒ (2) Suppose that G is 2-rankable. Since P_3 has the unique 2-ranking 121, every starting and ending vertex must get the same color 1 in G. Consequently, *every* path P (not only the alternating ones) joining two such vertices must have even length, for otherwise the endpoints of P should get distinct colors in every proper 2-coloring (not only in the 2-rankings) of G.

(2) ⇒ (3) Let G be a graph satisfying condition (2), and suppose on the contrary that some $W = x_1x_2 \dots x_{2t} \subset G$ is an alternating walk of odd length, $2t - 1$, where both x_1 and x_{2t} are (possibly identical) central vertices. By definition, there exist directed paths of length 2, $P' = u'v'z'$ and $P'' = u''v''z''$, with $v' = x_1$ and $v'' = x_{2t}$. Denoting $x_0 := z'$ and $x_{2t+1} := u''$, observe that $W^* := u''W^{-1}z' = x_{2t+1}x_{2t}x_{2t-1} \dots x_1x_0$ is an alternating walk of odd length $2t + 1$ from the starting vertex x_{2t+1} to the ending vertex x_0. Now condition (2) implies that W^* cannot be a path, i.e., $x_i = x_j$ holds for some $0 \le i < j \le 2t + 1$. Assuming that $j - i$ is as small as possible, we find i and j so that $C := x_ix_{i+1} \dots x_j$ is a cycle.

We distinguish between two simple cases, depending on the parity of $i - j$. If $i - j$ is even, then C is an *odd* cycle in which x_i is the middle vertex of a directed P_3, namely either $x_{i+1}x_ix_{j-1}$ or $x_{j-1}x_ix_{i+1}$. Thus, $C - x_i$ is an alternating path of odd length from the starting vertex of this P_3 to its ending vertex, which is a contradiction to condition (2). On the other hand, if $i - j$ is odd, then removing the segment $x_{j-1}x_{j-2} \dots x_{i+2}x_{i+1}$ from W^* we obtain a shorter alternating walk of odd length from x_{2t+1} to x_0, and repeating the same argument we eventually get a final contradiction.

(3) ⇒ (4) Let G be a connected graph satisfying condition (3). We first show that G is bipartite. Suppose on the contrary that $C = x_1x_2 \dots x_{2k+1}$ is a cycle of odd length in G. By the assumption on parity, at least two consecutive edges are oriented in the same direction, and thus at least one vertex of C is central. It follows that, taking subscript addition modulo $2k + 1$, there exist two subscripts i and j (possibly $j = i+2k+1$) such that $j-i$ is odd, both x_i and x_j are central vertices, and no vertex x_k, $i < k < j$, is central. Then the walk $x_ix_{i+1} \dots x_j$ (or its inverse, $x_jx_{j-1} \dots x_i$) is alternating.

Next, we show that all central vertices are located in the same bipartition class of G. If this is not the case, let x, y be central vertices belonging to distinct classes and being at minimum distance apart. (Recall that G is connected.) Now, any shortest x–y path has odd length and is alternating, for otherwise G would contain two central vertices in distinct classes closer to each other than x and y.

(4) ⇒ (1) Let $V(G) = A \cup B$ be a bipartition of G such that all central vertices belong to A. Then the mapping that assigns 1 to the vertices in B and 2 to the vertices in A is a 2-ranking of G. ☐

Remarks. 1. Algorithmically it is very easy to decide whether a digraph G 1. Algorithmically it is very easy to decide whether a digraph G is 2-rankable. Indeed, the answer is negative whenever G is not bipartite, and otherwise it suffices to test separately in each connected component if some of the two possible 2-colorings are 2-rankings. See also condition (4) in Theorem 5.1.

2. Similar types of problems have been studied in the framework of precoloring extension in several papers. Good characterizations are known for the existence of k-colorings of trees with any number of prescribed monochromatic independent sets [2, 3], and also for one prescribed monochromatic independent set in *perfect* graphs [7]. (As we have mentioned before, the problem for bipartite graphs with at least three precolored vertices of distinct colors is algorithmically hard [6], and this is true for two monochromatic vertex pairs in distinct colors, too.) For an extensive survey on this subject, see [12].

3. Some small subgraphs excluded by the degenerate "alternating" path of length 1 are as follows:

- The cyclic triangle $y_1 \to y_2 \to y_3 \to y_1$, where any two of the y_i are adjacent central vertices and also each edge joins a starting vertex with an ending vertex;
- the transitive triangle $y_1 \to y_2 \to y_3 \leftarrow y_1$, where y_1y_3 is an edge from a starting vertex to an ending vertex (and $y_2y_3y_1y_2$ is an odd alternating walk from the central vertex y_2 to itself);
- the path $y_1 \to y_2 \to y_3 \to y_4$ of length 3, where the edge y_2y_3 joins a starting vertex with an ending vertex, both of which are central as well.

Moreover, chordless odd cycles of lengths ≥ 5 (with any orientation) are also excluded by the longer alternating paths or by the entire cycle as an alternating walk, according to conditions (2) and (3) for longer paths and walks. Note that the characterization of 2-rankable digraphs in terms of forbidden subgraphs involves an *infinite* family of minimal configurations, which is not the case for undirected rankings.

6. Open problems.

There are many interesting related problems arising in the above context in a natural way. Below we mention some of them.

(1) Draw a sharper line between the polynomial instances of oriented trees (and oriented planar graphs of bounded treewidth) and the NP-complete class of directed acyclic bipartite planar graphs, by describing large subclasses of the latter in which the ranking number still can be determined in polynomial time.

(2) What is the complexity of Directed Edge Ranking for a fixed number of colors? (The undirected version is linear [1] but NP-complete if the number of colors is unrestricted [8].)

(3) More generally, which classes of directed graphs admit polynomial-time decision algorithms for k-ranking or edge k-ranking or both, for every fixed k?

Acknowledgment. The second author thankfully acknowledges support from the Konrad-Zuse-Zentrum für Informationstechnik Berlin, where part of this research was carried out.

REFERENCES

[1] H. L. Bodlaender, J. S. Deogun, K. Jansen, T. Kloks, D. Kratsch, H. Müller, and Zs. Tuza, *Rankings of graphs*, SIAM J. Discrete Math., 11 (1998), pp. 168–181.

[2] M. Hujter and Zs. Tuza, *Precoloring extension.* II. *Graph classes related to bipartite graphs*, Acta Math. Univ. Comenian., 62 (1993), pp. 1–11.

[3] M. Hujter and Zs. Tuza, *Precoloring extension.* III. *Classes of perfect graphs*, Combin. Probab. Comput., 5 (1996), pp. 35–56.

[4] A. V. Iyer, H. D. Ratliff, and G. Vijayan, *Parallel Assembly of Modular Products—An Analysis*, Technical Report 88-06, PDRC, Georgia Institute of Technology, Atlanta, GA, 1988.

[5] A. V. Kostochka, E. Sopena, and X. Zhu, *Acyclic and oriented chromatic numbers of graphs*, J. Graph Theory, 24 (1997), pp. 331–340.

[6] J. Kratochvíl, *Precoloring extension with fixed color bound*, Acta Math. Univ. Comenian., 62 (1993), pp. 139–153.

[7] J. Kratochvíl and A. Sebő, *Coloring precolored perfect graphs*, J. Graph Theory, 25 (1995), pp. 207–215.

[8] T. W. Lam and F. L. Yue, *Edge ranking of graphs is hard*, Discrete Appl. Math., 85 (1998), pp. 71–86.

[9] A. Raspaud and E. Sopena, *Good and semi-strong colorings of oriented planar graphs*, Inform. Process. Lett., 51 (1994), pp. 171–174.

[10] A. Sali and G. Simonyi, *Oriented list colorings of undirected graphs*, in Contemporary Trends in Discrete Mathematics, R. L. Graham, J. Kratochvíl, J. Nešetřil, and F. S. Roberts, eds., DIMACS Ser. Discrete Math. Theoret. Comput. Sci., AMS, Providence, RI, 1999, pp. 307-316.

[11] E. Sopena, *The chromatic number of oriented graphs*, J. Graph Theory, 25 (1997), pp. 191–205.

[12] Zs. Tuza, *Graph colorings with local constraints—a survey*, Discuss. Math. Graph Theory, 17 (1997), pp. 161–228.

[13] Zs. Tuza and M. Voigt, *Oriented List Colorings*, manuscript, 1998.

SIAM J. DISCRETE MATH.
Vol. 12, No. 3, pp. 385–411

ON-LINE 3-CHROMATIC GRAPHS I. TRIANGLE-FREE GRAPHS*

ANDRÁS GYÁRFÁS†, ZOLTÁN KIRÁLY‡, AND JENŐ LEHEL§

Abstract. This is the first half of a two-part paper devoted to on-line 3-colorable graphs. Here on-line 3-colorable triangle-free graphs are characterized by a finite list of forbidden induced subgraphs. The key role in our approach is played by the family of graphs which are both triangle- and $(2K_2 + K_1)$-free. Characterization of this family is given by introducing a bipartite modular decomposition concept. This decomposition, combined with the greedy algorithm, culminates in an on-line 3-coloring algorithm for this family. On the other hand, based on the characterization of this family, all 22 forbidden subgraphs of on-line 3-colorable triangle-free graphs are determined. As a corollary, we obtain the 10 forbidden subgraphs of on-line 3-colorable bipartite graphs. The forbidden subgraphs in the finite basis characterization are on-line 4-critical, i.e., they are on-line 4-chromatic but their proper induced subgraphs are on-line 3-colorable. The results of this paper are applied in the companion paper [Discrete Math., 177 (1997), pp. 99–122] to obtain the finite basis characterization of *connected* on-line 3-colorable graphs (with 51 4-critical subgraphs). However, perhaps surprisingly, connectivity (or the triangle-free property) is essential in a finite basis characterization: there are infinitely many on-line 4-critical graphs.

Key words. on-line coloring, forbidden subgraphs

AMS subject classifications. 05C15, 05C75, 05C85

PII. S089548019631030X

Introduction. A *proper coloring* of a graph G is an assignment of positive integers (called colors) to its vertices in such a way that adjacent vertices have distinct colors. The smallest number of colors in any proper coloring is denoted by $\chi(G)$ and is called the *chromatic number* of G. An *on-line coloring* of a (finite) G is an algorithm that colors the vertices as follows:

- Vertices of G are given in some order $v_1, v_2, \ldots$ (unknown by the algorithm).
- In the ith step the algorithm assigns a proper color to v_i (and never changes it later).

The most extensively studied on-line coloring algorithm is the greedy or *first fit* algorithm (FF): in each step it assigns the smallest available positive integer as color to the current vertex. In general, on-line coloring can be interpreted as a two-person game of GraphDrawer and GraphPainter. Drawer's moves consist of successively revealing vertices of a graph G with all adjacencies to vertices already known by Painter, and in each step Painter assigns a color to the current vertex. Painter's aim is to use as few distinct colors as possible while Drawer's aim is to force Painter to use as many colors as possible. The common optimum value will be called the on-line chromatic number of G.

*Received by the editors October 4, 1996; accepted for publication (in revised form) November 23, 1998; published electronically September 7, 1999.
http://www.siam.org/journals/sidma/12-3/31030.html

†Computer and Automation Research Institute, Hungarian Academy of Sciences, Kende u. 13-17, Budapest, H-1111, Hungary (gyarfas@luna.aszi.sztaki.hu). The work of this author was supported by OTKA grant T 16414.

‡Eötvös Loránd University, Department of Computer Science, Rákóczi út 5, Budapest, H-1088, Hungary (kiraly@cs.elte.hu). The work of this author was supported by OTKA grants F 014919, T 14302, and T 17580.

§Computer and Automation Research Institute, Hungarian Academy of Sciences, Kende u. 13-17, Budapest, H-1111, Hungary, and University of Louisville, Louisville, KY (j0lehe01@athena.louisville.edu). The work of this author was supported by OTKA grant T 16414.

Let G be a graph and A be some fixed on-line coloring algorithm. Then the maximum number of colors used by A during any coloring game (i.e., for all orderings of the vertices of G) is called the A-chromatic number of G and is denoted by $\chi_A(G)$. The *on-line chromatic number*, $\chi^*(G)$, is the minimum number of colors Painter succeeds with when playing on G; that is, $\chi^*(G) = \min\{\chi_A(G) : A \text{ is an on-line coloring}\}$. A graph G is (on-line) *k-critical* if $\chi^*(G) = k$ and $\chi^*(G') < k$ holds for every proper induced subgraph $G' \subset G$.

The concept of on-line chromatic number of graphs was introduced in [GL1], [GL2]; a similar notion, recursive coloring, had been investigated earlier. The introduction in [KPT1] gives a brief survey of the connection of these concepts. Our reference list covers several areas of on-line graph colorings beyond our particular subject [GKL2], [I], [K], [K1], [K2], [KK], [KT1], [KT], [LST], [V].

On-line 2-colorable graphs are rather trivial, and their connected components are complete bipartite graphs. This statement is a good introductory exercise to on-line colorings. It also shows that a single on-line algorithm, FF, provides a 2-coloring for every on-line 2-colorable graph. This is not the case for on-line 3-colorable graphs as demonstrated by the **B**-**E** paradigm [GL2]: although the graphs **B** and **E** (see Figure 1) are on-line 3-colorable, Painter cannot color with three colors if Drawer does not tell in advance which graph is to be presented. Thus a single on-line 3-coloring algorithm cannot 3-color every on-line 3-colorable graph. The same phenomenon explains that such a simple operation as addition of an isolated vertex may change on-line 3-colorability of a graph. The smallest amusing example is the triangle with a pendant edge on each of its vertices [GKL1]. A bipartite example comes from the evolution of **B**. Adding an isolated edge and an isolated vertex to the graph **B** gives an on-line 3-colorable graph, but if a further isolated vertex is added, an on-line 4-chromatic graph is obtained. These examples might suggest that on-line 3-colorable graphs are very restricted, but examples like the Petersen graph, $K_3 \times K_3$ [GKL1], seem to refute this view. It seems to us that the analysis of on-line 3-colorable graphs is a good test case by which to understand paradoxical features of on-line colorings. As pointed out by referees, our approach is tailored specifically to 3-colorable graphs and at many places relies heavily on case analysis. Unfortunately, this seems to be an inherent feature of the subject.

This paper gives a characterization of on-line 3-colorable triangle-free graphs. The crucial role is played by the family of graphs which are both triangle- and $(2K_2+K_1)$-free. We use the notation $(\mathbf{\Delta}, \mathbf{\Xi})$-free for this family in accordance with our notations $\mathbf{\Delta}$ for the triangle C_3 and $\mathbf{\Xi}$ for $2K_2 + K_1$. Our key result (Theorem 1) states that $(\mathbf{\Delta}, \mathbf{\Xi})$-free graphs are on-line 3-colorable—in fact, with a single on-line algorithm $\mathcal{A}$ (section 3).

Theorem 1 is related to coloring results on $(\mathbf{\Delta}, T)$-free graphs. A well-known conjecture [G], [S] states that $(\mathbf{\Delta}, T)$-free graphs have bounded chromatic number in terms of the number of vertices of T, where T is a forest. The on-line version behaves differently; in [GL1] it was shown that the on-line chromatic number of $(\mathbf{\Delta}, P_6)$-free graphs is not bounded. Sumner proved that $(\mathbf{\Delta}, P_5)$-free graphs are 3-colorable [S] and in fact are 3-colorable by FF as shown in [GL3]. A well-known example (the bipartite complement of mK_2) demonstrates that the FF-chromatic number is unbounded for our $(\mathbf{\Delta}, \mathbf{\Xi})$-free (even for $(\mathbf{\Delta}, K_2 + 2K_1)$-free) family. Thus the on-line 3-coloring algorithm $\mathcal{A}$ of Theorem 1 cannot be replaced by FF. Actually, $\mathcal{A}$ seems to be the first algorithm essentially different from FF which is optimal for a family where FF behaves very poorly. It is worth noting that, going a step further, the family of

$(\mathbf{\Delta}, 3K_2)$-free graphs are not 3-colorable even off-line since the Grötzsch graph is in the family. Finally we note that (prepared by works in [GL2], [GL3], [KPT]) a deep theorem of Kierstead, Penrice, and Trotter [KPT1] implies that the family of $(\mathbf{\Delta}, T)$-free graphs has a bounded on-line chromatic number if and only if each component of the forest T is P_6-free.

Structural and coloring properties of $(\mathbf{\Delta}, \Xi)$-free graphs are interrelated. On one hand, algorithm $\mathcal{A}$ is used to prove structural results; for example, the existence of $\mathcal{A}$ immediately implies (through the **B**-**E** paradigm) that $(\mathbf{\Delta}, \Xi)$-free graphs cannot contain both **B** and **E**. On the other hand, algorithm $\mathcal{A}$ is based on our structural characterization of the family.

To obtain a general structure theorem (Theorems II and 2) we shall introduce a modular decomposition of Ξ-free bipartite graphs in section 2. The building blocks (modules) are $2K_2$-free bipartite graphs (halfgraphs), and they are joined using complete bipartite graphs. Nonbipartite members of the family are obtained by extending bipartite ones having at most two modules, and their structure shows a peculiar circular symmetry (Theorem 1 and (2.7)). This is a graph theoretic structure theorem independent of on-line coloring and so has its own interest.

In section 4.2 we extend algorithm $\mathcal{A}$ to color disconnected $\mathbf{\Delta}$-free graphs containing **B** with three colors when it is possible.

A synthesis of our techniques results in a characterization of on-line 3-colorable triangle-free graphs by finitely many (22) forbidden subgraphs (Theorem 4). In fact, these are the triangle-free on-line 4-critical graphs displayed in Figures 3, 4, and 5 (except F_1 and F_5). We have learned that the Drawer-Painter game is rather interesting on almost all of them due to diverse strategies with subtle details. During a game on any of these graphs, a smart Painter has a chance to achieve a 3-coloring against an imperfect Drawer. However, a perfect Drawer can always force any Painter to use four colors.

Theorem 4 implies that on-line 3-colorability of a triangle-free graph can be decided (theoretically) in polynomial time of its order, in contrast with off-line 3-colorability which is known to be NP-complete [L].

In the companion paper [GKL1] Theorems 2 and 3 were used to obtain the finite basis characterization of *connected* on-line 3-colorable graphs (with 51 forbidden on-line 4-critical subgraphs). In contrast to our expectations, the assumption of connectivity was essential: we found an infinite family of (disconnected) on-line 4-critical graphs. Therefore, on-line 3-colorable graphs (like off-line 2-colorable, i.e., bipartite graphs) cannot be characterized with finitely many forbidden subgraphs.

We conclude the introduction with remarks concerning algorithmic aspects of our results. The structural properties of on-line 3-colorable graphs developed in this paper and in its companion led to a very simple on-line coloring algorithm (FF(C_6) in [GKL1]). This algorithm is a slight modification of FF, easy to implement, and uses at most four colors on every on-line 3-colorable graph. Due to the **B**-**E** paradigm, this is the best that a single on-line algorithm can achieve. Another algorithm for the same purpose, List First Fit, was found independently by Kolossa [KO]. Vaguely speaking, both algorithms are fast optimal, but it is extremely difficult to prove that they do what they claim. Our attempt to sacrifice accuracy for clarity and the hope of generalization led to an on-line algorithm for which it is easy to bound the maximum number of colors (142) for any on-line 3-colorable input graph. Unfortunately, for $k > 3$, the proof is not suitable to give an affirmative answer for the following more general and seemingly important question. For fixed k, is it possible to find a single

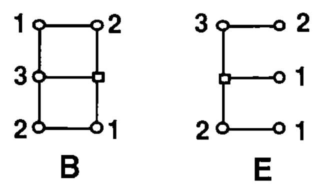


FIG. 1. *Graphs* **B** *and* **E**.

on-line coloring algorithm A_k which colors every on-line k-colorable graph with a bounded number of colors (in terms of k)? [GKL3].

1. Notations and results. Let K_n, P_n, and C_n denote the n-clique, the induced path with n vertices, and the induced n-cycle, respectively. For a positive integer k, kG is the union of k disjoint copies of G and $G+H$ is the disjoint union of the graphs G and H. We use the following nonstandard notation: $\mathbf{II} = 2K_2$, $\mathbf{\Xi} = \mathbf{II} + K_1$, **B** is a 6-cycle together with a long chord, and **E** is the graph obtained from **B** by removing two consecutive edges from its 6-cycle adjacent to the long chord (see Figure 1). The triangle is often denoted by $\mathbf{\Delta}$. Graphs with more than one forbidden subgraph are indicated by the list of subgraphs within parentheses.

The main result of the paper is the following theorem.

THEOREM 1. *If G is $(\mathbf{\Delta}, \mathbf{\Xi})$-free, then $\chi^*(G) \leq 3$. In addition, a 3-coloring for all $(\mathbf{\Delta}, \mathbf{\Xi})$-free graphs is obtained by a single on-line algorithm, $\mathcal{A}$.*

The proof of Theorem 1 is presented in sections 2 and 3. In section 2 we prove a structure theorem for $(\mathbf{\Delta}, \mathbf{\Xi})$-free graphs by introducing a new modular decomposition concept. Section 3 concludes the proof of Theorem 1 by presenting the on-line 3-coloring algorithm $\mathcal{A}$, a combination of FF and a natural but not simple algorithm based on the structure theorem.

Structural characterization of $(\mathbf{\Delta}, \mathbf{\Xi})$-free graphs is developed in several stages. First, **II**-free members of the family are described (see (2.2) and (2.3)). Next, bipartite $\mathbf{\Xi}$-free graphs are characterized using a modular decomposition technique. The decomposition relies on the fact that a bipartite graph G is $\mathbf{\Xi}$-free if and only if every connected component of the bipartite complement of G contains no **II** (see (2.4)). Finally we give extension rules by which all nonbipartite members of the family are derived from bipartite ones (see (2.6) and (2.7)). We summarize here the conclusion of section 2 without explaining the definitions in details. (These can be found at the end of the present section and throughout section 2.)

THEOREM I. *A $\mathbf{\Delta}$-free graph G with no equivalent vertices is $\mathbf{\Xi}$-free if and only if G satisfies one of the following properties:*

(a) $G = C_5 + K_1$.

(b) *G is a bipartite graph such that its bipartite complement is the disjoint union of connected* **II***-free bipartite graphs (called reduced halfgraphs).*

(c) *G is the induced subgraph of a graph H with the following structure: The vertices of H are partitioned into six nonempty sets $A_{i,j}, 1 \leq i \leq 3, 1 \leq j \leq 2$, such that the graph induced by A_{i_1,j_1} and A_{i_2,j_2} is a complete bipartite graph, if $i_1 = i_2, j_1 \neq j_2$; a halfgraph or a reduced halfgraph, if $i_1 \neq i_2, j_1 = j_2$; and a graph with no edges otherwise. Furthermore, for any $x \in A_{1,j}, y \in A_{2,j}, z \in A_{3,j}$ the set $\{x, y, z\}$ induces neither a triangle nor the complement of a triangle.*

The coloring result of Theorem 1 leads to the following theorem.

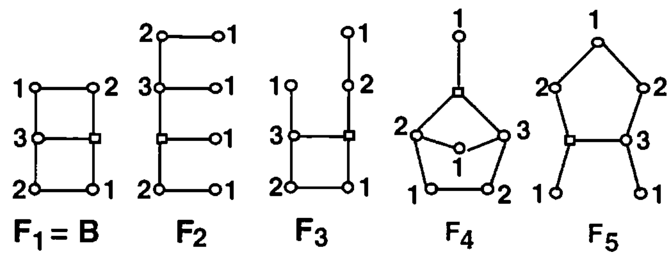


FIG. 2. *The five minimal* $\mathbf{\Delta}$*-free graphs of* FF*-chromatic number* 4.

THEOREM II. *Let* G *be a connected* $\mathbf{\Delta}$*-free graph containing a copy of* $\mathbf{B}$*. Then* G *is* $\mathbf{E}$*-free if and only if* G *is* Ξ*-free.*

For this purely graph theory statement we could not find a short direct proof that avoids on-line colorings. Actually, Theorem II is proved in the following stronger form in section 3.1.

THEOREM 2. *If* G *is a connected* $\mathbf{\Delta}$*-free graph containing a copy of* $\mathbf{B}$*, then the following statements are equivalent:*

(1) G *is* $\mathbf{E}$*-free.*
(2) G *has no induced subgraph isomorphic to any of* F_3, F_4 *in Figure* 2 *and* B_1, B_2, B_3, B_4 *in Figure* 4.
(3) G *is* Ξ*-free.*
(4) G *has on-line chromatic number* $\chi^*(G) \leq 3$.

Theorem 2 also helps in finding the list of all minimal graphs that are excluded from graphs of on-line chromatic number 3. Before formulating this result in Theorem 3 we present some critical graphs from the list. Let us start with the observation that any graph G of on-line chromatic number 4 must contain an induced subgraph G' such that $\chi_{FF}(G') = 4$. In [GKL1] we determined all graphs with FF-chromatic number 4 which are minimal with respect to that property. From the list of these 22 graphs, Figure 2 shows the $\mathbf{\Delta}$-free ones.

In [GKL1] it was also shown that F_2, F_3, and F_4 are 4-critical, $F_1 = \mathbf{B}$ and F_5 are not. Hence, if G is a $\mathbf{\Delta}$-free 4-critical graph different from F_2, F_3, and F_4, then G contains at least one of $\mathbf{B}$ and F_5. Figure 3 shows all 4-critical graphs obtained in [GKL1] which contain F_5.

The analysis of 4-critical graphs results in the following finite basis theorems.

THEOREM 3. *If* G *is a* $\mathbf{\Delta}$*-free graph containing* $\mathbf{B}$*, then* G *has on-line chromatic number at most* 3 *if and only if* G *has no induced subgraph isomorphic to any of* F_3, F_4 *in Figure* 2 *and* B_i*,* $1 \leq i \leq 10$*, in Figure* 4.

THEOREM 4. *A* $\mathbf{\Delta}$*-free graph* G *has on-line chromatic number* 3 *if and only if* G *has no induced subgraph isomorphic to any of* F_2, F_3, F_4 *in Figure* 2 *and the* 19 *graphs in Figures* 3 *and* 4.

A corollary of Theorem 4 is the following finite basis result for bipartite graphs.

THEOREM 5. *A bipartite graph* G *has on-line chromatic number at most* 3 *if and only if* G *has no induced subgraph isomorphic to* F_2, F_3 *in Figure* 2 *and* $B_1, B_2, B_3, B_5, B_7, B_8, B_9, B_{10}$ *in Figure* 4.

The vertex and the edge set of a graph G is respectively denoted by $V(G)$ and $E(G)$. The relation $D \subset G$ means that D is an induced subgraph of G. Throughout the paper *subgraph* always means *induced subgraph* (i.e., "G has a P_4" actually means that P_4 is an induced subgraph of G). For $D \subset G$ and $v \in V(G)$, $D + v$ and $D - v$

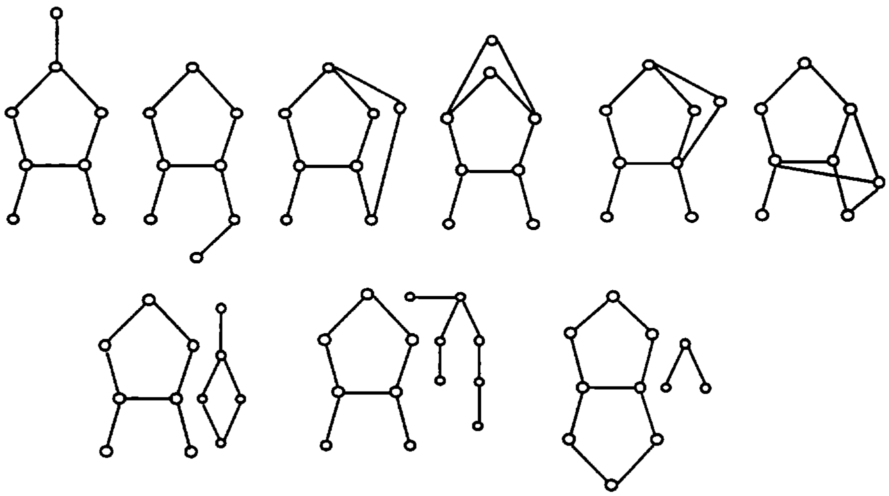


FIG. 3. *All* $\boldsymbol{\Delta}$*-free* 4*-critical graphs containing* F_5.

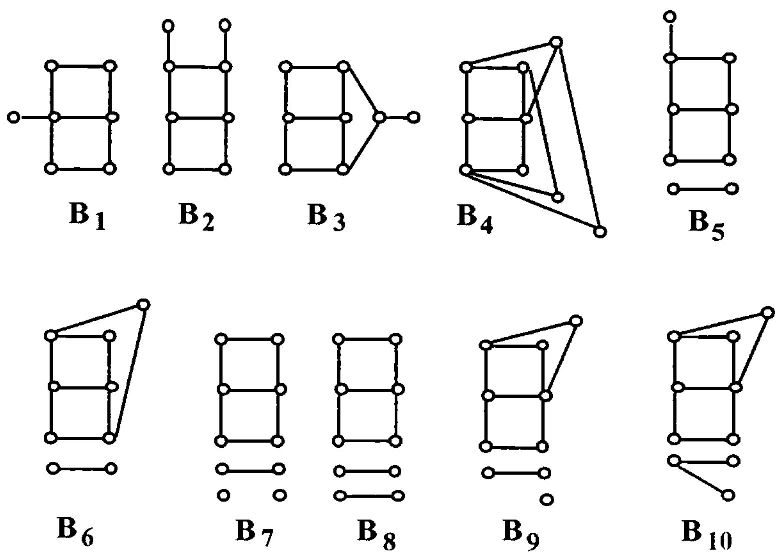


FIG. 4. *All* $\boldsymbol{\Delta}$*-free* 4*-critical graphs containing* $\mathbf{B}$.

denote the subgraph of G induced by $V(D) \cup \{v\}$ and $V(D) \setminus \{v\}$, respectively.

Define $N(v) = \{u \in V(G) : uv \in E(G)\}$ (and $N_U(v) = \{u \in U \subseteq V(G) : uv \in E(G)\}$) to be the ($U$-) neighborhood set of $v \in V(G)$. Vertices $u, v \in V(G)$ are called *equivalent* if and only if $N(u) = N(v)$. A graph is called *primitive* if it contains no equivalent pair of vertices. Vertex multiplication is the operation of replacing a vertex x of a graph with a certain number of equivalent copies of x. If a graph G is the vertex multiplication of some primitive graph G' then we say that G' is a *primitive representative* of G. For a graph G and $v \in V(G)$, let $\mathcal{C}_G(v) \subset V(G)$ denote the set of all vertices of G equivalent to v. Obviously, any subgraph of G induced by the set containing one vertex from each equivalence class is a primitive representative of G.

Throughout the paper it is assumed that bipartite graphs are given together with a bipartition of their vertices. A bipartite graph with partite sets X and Y is denoted by $[X, Y]$ and is called here a *bigraph*. General graphic operations have natural bipartite versions for bigraphs. In case of bigraphs the equivalence and the vertex multiplication are involving vertices in the same partite set. A bigraph is called primitive if it contains no equivalent pair of vertices. (A primitive bigraph can have two isolated vertices, one in each class.) For a bigraph $G = [X, Y]$, let $\widehat{G}$ denote the bipartite complement of G, that is, $x \in X$ and $y \in Y$ are adjacent in $\widehat{G}$ if and only if xy is not an edge of G. A vertex v is called a *star vertex* of the bigraph $[X, Y]$ if $N(v) = X$ or Y. A subgraph of an arbitrary graph G induced by two disjoint independent sets $X, Y \subset V(G)$ is a bigraph $[X, Y]$ of G.

2. Characterizations of $(\mathbf{\Delta}, \mathbf{\Xi})$-free graphs. Graph classes we shall consider in this section are closed under vertex multiplication. (It is worth noting that this is not true for the whole class of on-line 3-colorable graphs.) This property is formulated in our first proposition. (The trivial proof is omitted.)

(2.1) Let H be a primitive (bi)graph. Then G is an H-free (bi)graph if and only if any primitive representative of G is H-free. □

A $\mathbf{\Xi}$-free graph G with no triangle is of one of the following two types: either G is **II**-free or G contains **II** and is disconnected (Type 1), or G contains **II** and is connected (Type 2).

Type 1. Let G be a graph of Type 1. No connected component of G may contain **II**; otherwise, G would be connected, which contradicts the definition of Type 1 graphs. If G has two nontrivial connected components, then it contains **II**; thus no third component might exist. Because both components must be $(K_2 + K_1)$-free, G is bipartite, and it is the disjoint union of two complete bipartite graphs. Assume next that G has exactly one nontrivial connected component, that is, G is **II**-free.

First let G be a nonbipartite graph of Type 1. Since G is $\mathbf{\Delta}$-free with no **II**, its shortest induced odd cycle must be a C_5. This observation combined with (2.1) results in the following easy characterization.

(2.2) A nonbipartite $\mathbf{\Delta}$-free graph G is of Type 1 if and only if G is the vertex multiplication of C_5 or $C_5 + K_1$. □

Next let G be a bipartite graph of Type 1 containing one nontrivial connected component, or, equivalently, let $G = [X, Y]$ be a **II**-free bigraph. The following four properties are obviously equivalent:

(i) $G = [X, Y]$ is **II**-free;
(ii) for every $x, x' \in X$, either $N(x) \subseteq N(x')$ or $N(x') \subseteq N(x)$;
(iii) X has an ordering $x_1, \dots, x_p$ such that $Y \supseteq N(x_1) \supseteq \cdots \supseteq N(x_p)$;
(iv) Y has an ordering $y_1, \dots, y_q$ such that $N(y_1) \subseteq \cdots \subseteq N(y_q) \subseteq X$.

The equivalence of (i) and (iii) characterizes **II**-free bigraphs as follows: $G = [X, Y]$ is **II**-free if and only if $\{N(x) : x \in X\}$ defines a chain on Y (and $\{N(y) : y \in Y\}$ defines a chain on X). The chain on X may start with the empty set (corresponding to an isolated vertex of Y); it may contain several copies of the same subset (corresponding to equivalent vertices of Y), and its last member is either the whole set X (which corresponds to a star vertex in Y) or the set of nonisolated vertices in X.

Using these observations together with (2.1), all **II**-free bigraphs can be obtained from the containment graphs of simple chains, called halfgraphs. The nth halfgraph,

$H(n)$, is defined as a bigraph on vertex set $\{x_1, \dots, x_n\} \cup \{y_1, \dots, y_n\}$ with $x_i y_j$ being an edge if and only if $i < j$. Notice the symmetry of $H(n)$ defined by the automorphism $x_i \longleftrightarrow y_{n+1-i}$ $(i = 1, \dots, n)$ between its partite sets. In this paper we call a bigraph halfgraph if it is a vertex multiplication of some $H(n)$. The primitive halfgraphs will be written as halfgraphs. A halfgraph which does not have isolated vertices in both bipartition classes will be called a *reduced halfgraph*.

(2.3) A bigraph G is of Type 1 if and only if G is the vertex multiplication of **II** or it is a halfgraph or a reduced halfgraph. □

The type of Ξ-free bigraphs can be determined by introducing a new modular decomposition concept which also will be useful for the whole structural description of $(\mathbf{\Delta}, \Xi)$-free graphs. Observe first that the bipartite complement of the bigraph Ξ is a P_5. Hence a bigraph G is Ξ-free if and only if its bipartite complement $\widehat{G}$ is P_5-free. Furthermore, a connected component of $\widehat{G}$ contains no P_5 if and only if it is **II**-free. According to the discussion before (1.3) each connected **II**-free bigraph is a connected reduced halfgraph, that is, either some isolated vertices or a halfgraph with all of its isolated vertices removed. Note that $\widehat{\mathbf{II}} \cong \mathbf{II}$; hence the connected components of $\widehat{\mathbf{II}}$ are isomorphic to K_2 (i.e., a connected reduced $H(2)$).

(2.4) A bigraph is Ξ-free if and only if every connected component of its bipartite complement is a connected reduced halfgraph. □

Let G be a Ξ-free bigraph and denote by $G_1, \dots, G_k$ the connected components of its bipartite complement $\widehat{G}$. Then $\widehat{G_i}$, $i = 1, \dots, k$, are called the *modules* of G. If a module contains just one vertex, then it is called a *trivial module*; otherwise, it is a *nontrivial module*. Observe that any two vertices from distinct partite sets and from distinct modules are adjacent in G; in particular, trivial modules are star vertices of the bigraph. It is easy to check that the bipartite complement of a connected reduced halfgraph is either a single vertex or a halfgraph. Therefore, by (2.3), *each nontrivial module of the unique module decomposition of G is a halfgraph.* Note that the two modules of the bigraph **II** are isomorphic to $H(1)$. Thus we obtain that a Ξ-free bigraph G is of Type 1 if and only if G has $k \leq 2$ nontrivial modules and, in case of $k = 2$, neither contains an edge.

Type 2. As a result of the module decomposition concept introduced for Ξ-free bigraphs we obtain that a bigraph G is of Type 2 if and only if G is connected and has $k \geq 2$ nontrivial modules. Nonbipartite graphs of Type 2 will be described as extensions of Ξ-free bigraphs.

Let G be a graph, $D \subset G$, and $z \in V(G) \setminus V(D)$. If D is a bigraph and $D + z$ is also bipartite, then z is called a *bipartite extension* of D. If $D + z$ is nonbipartite (i.e., z induces an odd cycle together with some vertices of D), then z is called an *odd extension* of D. The obvious transition rule of bipartite extensions are described as follows.

(2.5) Let G be a $(\mathbf{\Delta}, \Xi)$-free graph and let $D = [X, Y]$ be a connected induced bigraph of G. If $z \in V(G) \setminus V(D)$ is a bipartite extension of D and M is the module of $D + z$ containing z, then $M = \{z\}$, or $M - z$ is a module of D, or $M - z$ consists of at most one nontrivial module and a set of trivial modules of D. □

For characterizing nonbipartite graphs of Type 2 we need to extend the notion of halfgraphs. Let $F = [X, Y]$ be a halfgraph and let z be a new vertex adjacent to some vertices of F, i.e., z is an extension of F with neighborhood sets $X(z) \subseteq X$ and

$Y(z) \subseteq Y$. The graph $F + z$ is called an *extended halfgraph* if the following properties are all satisfied:

- $X(z) \neq \emptyset$ and $Y(z) \neq \emptyset$.
- $F + z$ is $\mathbf{\Delta}$-free.
- If $x \in X$, $y \in Y$, $xy \notin E$, then at least one of zx and zy is an edge.

The second and third properties together say that there are neither triangles nor empty triangles of form zxy.

The following statement describes the structure of nonbipartite graphs of Type 2.

(2.6) Let G be a connected $(\mathbf{\Delta}, \mathbf{\Xi})$-free graph and let $D = [X, Y]$ be an induced bigraph of G. Assume that the set $Z \subset V(G) \setminus V(D)$ of all odd extensions of D is nonempty. Then D and Z satisfy (i)–(iv):

(i) D has at most two nontrivial modules. Furthermore, for any fixed $z \in Z$, the neighbors of z in D belong to the same nontrivial module.

(ii) If $M_1 \subset D$ is the module containing the neighbors of $z \in Z$, then $M_1 + z$ is an extended halfgraph.

(iii) If $z_1, z_2 \in Z$ are distinct, then $z_1 z_2 \in E(G)$ if and only if z_1 and z_2 are adjacent to distinct nontrivial modules of D.

(iv) Let $M_1 = [X_1, Y_1] \subset D$ be a module and $Z_1 = \{z \in Z$: neighbors of z are in $M_1\}$, and let A, B, C denote the sets X_1, Y_1, Z_1 in any order and $c_1, c_2 \in C$. Then either $N_A(c_1) \subseteq N_A(c_2)$ or $N_A(c_2) \subseteq N_A(c_1)$. Moreover, if $N_A(c_1) \subset N_A(c_2)$, then $N_B(c_2) \subseteq N_B(c_1)$.

Proof. Recall that every nontrivial module of D is a halfgraph and its trivial modules are star vertices. Because G is $\mathbf{\Xi}$-free with no triangle, all induced odd cycles of $D + z \subset G$ are isomorphic to either C_5 or C_7. Because z is an odd extension, at least one induced odd cycle containing z must exist in $D + z$.

Because G is connected, every nonbipartite subgraph of G must be connected (otherwise, as easily can be checked, G would contain $\mathbf{\Xi}$). In particular, $D + z$ is connected for every $z \in Z$.

(i) Let $z \in Z$ and assume that C is an induced odd cycle of $D + z$ with $x \in V(C) \cap X(z)$ and $y \in V(C) \cap Y(z)$. Because $xy \notin E(G)$, x and y are vertices of the same module, say, $M_1 = [X_1, Y_1]$. If $u \in V(D)$ is a vertex not in M_1, then one of ux and uy is an edge of G; thus $zu \notin E(G)$ follows (because G is $\mathbf{\Delta}$-free). This shows that the neighbors of z in D belong to M_1. Assuming that D has more than two nontrivial modules, a copy of $\mathbf{II}$ between M_2 and M_3 together with z would induce a $\mathbf{\Xi}$ of G. Thus D has at most two nontrivial modules.

(ii) Let M_1, x, and y be as in case (i). As G is $\mathbf{\Delta}$-free and z is an odd extension with all neighbors in M_1, $M_1 + z$ satisfies the first two properties of extended halfgraphs. Suppose there are $x' \in X_1$ and $y' \in Y_1$ such that none of $zx', x'y', y'z$ is an edge. First observe that as $xy \notin E$, one of xy' and $x'y$ is also not an edge (M_1 does not contain $\mathbf{II}$); by symmetry we can assume that $x'y \notin E$. As we noted at the beginning of the proof, the graph induced by $C + x'$ is connected. Denote the neighbor of x' in C by y^*. As $x'y^*$ is an edge, y^* differs from z, y, y' and is in Y. zy^* is not an edge because C was induced. Therefore, $x'y^*, zy$ and y' induce $\mathbf{\Xi}$, a contradiction.

(iii) Let $z_1, z_2 \in Z$ be vertices with neighbors in the same module $M_1 = [X_1, Y_1]$. For proving $z_1 z_2 \notin E$ it is enough to see that they have a common neighbor in M_1. Let $y_1 \in Y_1$ be an isolated vertex of M_1. Either it is a common neighbor and we are done, or, e.g., $z_1 y_1 \notin E$. Then by (ii) and by the third property of extended halfgraphs

z_1 is connected to every vertex in X_1 and by the first property z_2 is connected to at least one vertex in X_1.

Let $z_1z_2 \notin E(G)$, let $x_i \in X, y_i \in Y$ be neighbors of z_i in M_i for $i = 1, 2$, and suppose $M_1 \neq M_2$. Observe that the subgraph D' induced by $\{z_1, x_1, y_2, z_2, x_2, y_1\}$ is a C_6. Since $\widehat{C_6} = 3K_2$, D' has three nontrivial modules. Hence, by (i), $D' \subset G$ has no odd extensions, which contradicts $M_1 \neq M_2$.

(iv) The first statement says that none of the bigraphs $[A, B], [B, C]$, and $[C, A]$ contains **II**. As we know this fact about $[X_1, Y_1]$ it is enough to prove it for $[Z, X_1]$. Suppose z_1x_1 and z_2x_2 induce a **II**. If X has a vertex connected to neither z_1 nor z_2, then G contains $\boldsymbol{\Xi}$. Hence $X = X_1$. Now if a vertex in Y_1 is isolated in M_1, then it is also isolated in D. Since the vertices z_i are odd extensions, both of them must have a neighbor in Y_1 which is not isolated in M_1. However, in the halfgraph M_1 there exists an $x \in X_1$ which is connected to every nonisolated vertex of Y_1, so x cannot be connected to any of the z_i's.

To prove the second statement indirectly, suppose that $a \in N_A(c_2) \setminus N_A(c_1)$ and $b \in N_B(c_2) \setminus N_B(c_1)$. Since ac_2b is not a triangle, $ab \notin E$. Since ac_1b is not an empty triangle, $ab \in E$. □

In the next proposition we formulate a converse of (2.6) which shows that a graph with properties (i)–(iv) is $\boldsymbol{\Xi}$-free. To get an even nicer symmetry, we swap the role of X_1 and Y_1. The proof is routine and the details are left to the reader.

(2.7) Suppose that the vertices of a graph G are partitioned into six nonempty sets $A_{i,j}, 1 \leq i \leq 3, 1 \leq j \leq 2$, such that the graph induced by A_{i_1,j_1} and A_{i_2,j_2} is a complete bipartite graph if $i_1 = i_2, j_1 \neq j_2$; a halfgraph or reduced halfgraph if $i_1 \neq i_2, j_1 = j_2$; and a graph with no edges otherwise. Suppose furthermore that for any $x \in A_{1,j}, y \in A_{2,j}, z \in A_{3,j}$ the set $\{x, y, z\}$ induces neither a triangle nor the complement of a triangle. Then G is $\boldsymbol{\Xi}$-free. □

3. On-line 3-coloring of ($\boldsymbol{\Delta}, \boldsymbol{\Xi}$)-free graphs. Let G be $(\boldsymbol{\Delta}, \boldsymbol{\Xi})$-free graph. If G is of Type 1, then by (2.2) and (2.3) it is either bipartite or 3-chromatic. Assume now that G is of Type 2 and nonbipartite. Then it has the structure described in (2.6). In particular, there is a bipartite subgraph $[X, Y]$ with nontrivial modules $M_1 = [X_1, Y_1]$ and $M_2 = [X_2, Y_2]$ such that all odd extensions can be partitioned into sets Z_1 and Z_2 in a manner that a vertex in Z_i has neighbors only in Z_{3-i} and in M_i. Since the three sets X, $Z_1 \cup (Y \setminus Y_1)$ and $Z_2 \cup Y_1$ are all independent, we get the following result:

(3.1) If G is a $(\boldsymbol{\Delta}, \boldsymbol{\Xi})$-free graph, then $\chi(G) \leq 3$. □

This section contains the proof of Theorem 1 (stated in the introduction), which claims that the stronger $\chi^*(G) \leq 3$ also holds in (3.1). Let us consider the on-line coloring game on graph G. At some step of the game let $D \subset G$ denote the colored subgraph (i.e., the subgraph induced by the set of all colored vertices of G), and denote by z the current vertex to be colored. For any on-line coloring algorithm A and for an integer r, let $A(r)$ denote the set of all vertices of G colored with r. If x is a colored vertex, $c(x)$ will denote its color.

Our on-line algorithm $\mathcal{A}$ consists of three consecutive stages. In the first stage, called *FF-stage*, first fit coloring is applied. The FF-stage ends up when **II** first appears in $D + z$. The current vertex z that terminates the FF-stage gets a color in the second stage including a single step, called **II**-*step*. After a suitable color is assigned to z a bigraph $D_0 \subseteq D + z$, called *reference graph*, is defined in the **II**-step to start the last stage. In the *last stage* z is considered as the (bipartite or odd) extension of the actual reference graph D_0. In each step of the last stage the color of z is determined with respect to D_0 and D_0 is updated for the next step.

FF-stage. z gets the smallest color r such that z has no neighbor in D colored with r.

Note that FF assigns the same color to equivalent vertices. In the early steps of the coloring game D is bipartite and eventually is disconnected. In this case we assume, for convenience, that all isolated vertices of $D = [X, Y]$ belong to Y. In particular, $X \neq \emptyset$ implies that $A(2) \cap X \neq \emptyset$. Notice, however, that the partite set of an isolated vertex is undefined in D, that is, it may change at a subsequent step of the game.

(3.2) If G is a $(\mathbf{\Delta},\mathbf{II})$-free graph, then $\chi_{FF}(G) \leq 3$.

Proof. If G is nonbipartite, then by (2.2) its primitive representative is C_5 or $C_5 + K_1$. Since, in both cases, the maximum degree is 2, $\chi_{FF}(G) \leq 3$ follows. Note that the coloring of C_5 by FF is unique: 12123 (in some cyclic ordering of the vertices).

Assume now that G is bipartite and contains at least one edge. Now G is a reduced halfgraph. Recall that all isolated vertices are considered to be in Y. In any FF-coloring of G, by definition, FF(1) is a maximal independent set, and FF(2) is a maximal independent set in $G - \text{FF}(1)$. So either FF(1)=Y and FF(2)=X or FF(1) is a maximal independent set containing vertices from both X and Y and FF(2)$\subseteq X$, FF(3)$\subseteq Y$ such that each 2-colored vertex is connected to every 3-colored vertex (because a reduced halfgraph minus a maximal independent set is either a graph with no edges or a complete bipartite graph). □

The properties of the coloring patterns obtained in the proof of (3.2) will be used in the **II**-step below.

II-step. We shall see that starting with this step $\mathcal{A}$ is able to color the current vertex z so that the overall colored graph satisfies a set of properties we call Ruleset.

Ruleset for a graph D. In D there is maximal induced bigraph $D_0 = [X, Y]$ with nontrivial modules $M_1 = [X_1, Y_1]$, $M_2 = [X_2, Y_2]$, etc., and with some trivial modules such that all odd extensions of D_0 are connected to either M_1 (forming the set Z_1) or M_2 (forming the set Z_2). Furthermore, the coloring by $\mathcal{A}$ satisfies the following rules:

For some permutation s_1, s_2, s_3 of colors 1, 2, and 3,

(i) if x and y are equivalent vertices in D, then $c(x) = c(y)$;
(ii) $A(s_3) \cap Y \subseteq Y_1 \subseteq Y \subseteq A(s_1) \cup A(s_3)$;
(iii) $A(s_3) \cap X \subseteq X_1 \subseteq X \subseteq A(s_2) \cup A(s_3)$;
(iv) $Z_2 \subseteq A(s_3)$, $Z_1 \subseteq A(s_1) \cup A(s_2)$;
(v) the bigraphs $[A(s_1) \cap Y_1, A(s_2) \cap X_1]$, $[A(s_2) \cap Z_1, A(s_3) \cap Y_1]$, and $[A(s_1) \cap Z_1, A(s_3) \cap X_1]$ are complete.

In the **II**-step $\mathcal{A}$ determines D_0 and properly colors z so that $D + z$ satisfies Ruleset. Note that as the colored graph extends, the same Ruleset will be maintained by $\mathcal{A}$ in each step of the last stage. Now we show how Ruleset can be achieved in the **II**-step.

Take any **II** in $D + z$ and let $D_0 = [X_0, Y_0]$ be a maximal bipartite subgraph of $D + z$ containing that **II**. Clearly z is in D_0. As $D_0 - z$ is **II**-free, D_0 has exactly two nontrivial modules $M = [X, Y]$ and $M' = [X', Y']$ and we can assume $\{z\} = X$. Let y be an arbitrary vertex in Y.

Case 1. D is bipartite. Only M can have odd extensions forming the set Z.

If the graph D is colored by two colors, then let $s_1 = c(y)$, $s_2 = 3 - s_1$, $s_3 = 3$, and color z by 3. If D has no isolated vertices, then D is connected so the color of a vertex is uniquely determined by its distance (in D) from y and Ruleset is satisfied. Suppose D has some isolated vertices, those isolated vertices are in Y'. (They are connected to z because $D + z$ is $\mathbf{\Xi}$-free and they cannot be odd extensions.) If $c(y) = s_1 = 1$, then Ruleset remains satisfied. If $c(y) = s_1 = 2$, then change s_2 to 3 and s_3 to 1. Now with $M_1 = M'$ Ruleset is satisfied again. ($X' \subseteq A(1) = A(s_3)$, $Z \subseteq A(1) = A(s_3)$ and there cannot be trivial modules in X_0.)

If D is 3-colored, then first suppose that if $t \in Y$, then $c(t) \neq 1$. Thus all vertices in Z have color 1. Let $s_1 = c(y)$, $s_2 = 5 - s_1$, $s_3 = 1$, and color z by s_2. Note that in $(D_0 - z) \cup Z$ there is a complete bipartite graph between the s_1- and the s_2-colored vertices. Consequently every s_2-colored vertex which is different from z is connected to y. As vertices in Z have color s_3 all vertices with color s_2 are in X_0. Thus color s_2 for z is permitted and all vertices with color s_1 are in Y_0. The isolated vertices of D are in Y' as before so if none of the trivial modules is colored by $s_3 = 1$, then Ruleset is satisfied with $M_1 = M'$. As D is 3-colored, color 1 appears in Y_0, so trivial modules in X_0 have different color. If $y^* \in Y_0$ is a trivial module colored by 1, then X' is uniformly colored by s_2. In this case vertices in $Y_0 \setminus Y$ have no neighbors colored by 1, so they themselves are colored by $1 = s_3$ and consequently swapping s_1 and s_3 and choosing $M_1 = M$ Ruleset is satisfied again.

Now suppose D is 3-colored and there is a $t \in Y, c(t) = 1$. Observe that Y cannot be uniformly 1-colored because in this case color 3 could not appear in D, so there is a $t' \in Y, c(t') \neq 1$. Let $s_1 = 1$, $s_3 = c(t')$, $s_2 = 5 - s_3$. Now vertices in $X_0 - z$ are colored by s_2, so vertices in $Y_0 \setminus Y$ are colored by $1 = s_1$ and there is no s_2-colored vertex in Y. The s_3-colored t' has a 1-colored neighbor z_1; it must be in Z. If $z_2 \in Z$ is colored by s_3, then $z_1 t'$ and $z_2 t$ induce a $\mathbf{II}$, which is not the case. Thus we are allowed to color z by s_3. To check that Ruleset is satisfied with $M_1 = M$ we need to check rule (v). Every vertex in Z is connected to z. Let $z_2 \in Z$ be an arbitrary vertex colored by s_2. By (2.6)(iv) one of $N_Y(z_1)$ and $N_Y(z_2)$ contains the other. Clearly $N_Y(z_1) \subset N_Y(z_2)$ because z_1 does not have any 1-colored neighbor in Y while z_2 does. Therefore, the arbitrarily chosen s_2-colored $z_2 \in Z$ and s_3-colored $t' \in Y$ are connected.

Case 2. D is not bipartite. As $D+z$ is of Type 2, it is connected by the observation made at the beginning of the proof of (2.6)—every nonbipartite subgraph must be connected. Thus D is a vertex multiplication of C_5.

Let Z be the (maybe empty) set of odd extensions connected to M and Z' be the set of odd extensions connected to M'. As D is a vertex multiplication of C_5 both M and M' are vertex multiplications of H_1 and either Z is empty or there are no trivial modules in Y_0. Moreover, the equivalence classes of the C_5 are uniformly colored and one class is colored by 3 while the others are colored by 1212. It is easy to check (five cases depending on which class is colored by 3) that we can color z in a manner such that every $\mathbf{II}$ in $D + z$ will be colored by three colors and Ruleset is satisfied in all these cases with appropriate permutation s_1, s_2, s_3. □

Last stage. When the algorithm observes that D contains $\mathbf{II}$ it knows that D satisfies Ruleset and is able to determine appropriate D_0, M_1, M_2, and permutation s_1, s_2, s_3. In every step of the last stage $\mathcal{A}$ colors z in such a way that $D + z$ always satisfies Ruleset. In particular, G becomes 3-colored when $\mathcal{A}$ terminates.

Case 1. D is connected and z is an odd extension of D_0. If z is connected to M_2, then color z by s_3 and $D + z$ clearly satisfies Ruleset. Furthermore, suppose z is connected to M_1. If all neighbors of z are colored by s_3, then z is uniformly connected to either $A(s_3) \cap X_1$ or $A(s_3) \cap Y_1$; otherwise an empty triangle could be found. In the first case s_1 and in the second s_2 is the appropriate color for z to satisfy rule (v). Now by symmetry we can suppose that z is connected to an $x \in X_1$ such that $c(x) = s_2$. The following line of thought will be used in further cases:

(3.3) We claim that if $x' \in X_1$ and $c(x') = s_3$, then $zx' \in E$. For getting a contradiction suppose that z is connected to x but not to x'. By this assumption and by (2.6)(iv) $N_{Z_1+z}(x) \supset N_{Z_1+z}(x')$ in graph $D+z$ and so $N_{Y_1}(x) \subseteq N_{Y_1}(x')$. In the graph D all vertices in Z and Y_1 have a color satisfying rule (v), so $N_{Z_1}(x) = N_{Z_1}(x')$ and $N_{Y_1}(x) = N_{Y_1}(x')$. We get $x \sim x'$ in D and this contradicts rule (i).

To finish Case 1 observe that if z is connected to any $y \in Y_1$, $c(y) = s_1$, then xyz would be a triangle. The argument above says that z is connected to every s_3-colored vertex in X_1; consequently s_1 is the proper color for z.

Case 2. D is connected and z is a bipartite extension of D_0. Let $M = [X, Y]$ be the module of $D_0 + z$ which contains z. By symmetry we can assume $z \in X$. If $M \not\supseteq M_1$, then color z by s_2 and Ruleset remains satisfied. Suppose $M_1 \subseteq M$. First observe that vertices in $M \setminus M_1$ different from z cannot make any rule wrong. (They are trivial modules of D_0.) If z is equivalent to some vertex in M, then the color of that vertex is also good for z.

Suppose there is a $y \in Y$, $c(y) = s_3$, and $zy \in E$. If $z_1 \in Z_1$ and z is connected to z_1, then y is not connected to z_1 so $c(z_1) \neq s_2$. A similar argument as in (3.3) shows that z must be connected to every s_1-colored vertex in Y; consequently s_2 is a proper color for z.

The remaining case is that all neighbors of z in Y are colored by s_1. If z is uniformly connected to $A(s_1) \cap Z_1$, then s_3 is a proper color for z. If there is a $z_1 \in Z_1$ such that $c(z_1) = s_1$ and zz_1 is not an edge, then the absence of empty triangles shows that every s_1-colored vertex in Y is connected to z. An argument similar to (3.3) says that z has no s_2-colored neighbors in Z_1, so s_2 is the proper color for z.

Case 3. D is not connected. Now D is a vertex multiplication of **II** and is 3-colored (satisfying Ruleset). Note that when adding z to D some vertices might change their partite sets. To resolve this problem consider $D + z$ that is a connected bigraph (with a unique bipartition). Then remove z and keep the eventually modified bipartition for D. It is easy to check that in each case a permutation s_1, s_2, s_3 can be obtained so that Ruleset holds true for the modified bigraph D. Then the procedure described in Case 2 applies. This concludes the proof of Theorem 1.

4. $\boldsymbol{\Delta}$-free critical graphs of on-line chromatic number 4. In this section we characterize $\boldsymbol{\Delta}$-free 4-critical graphs. Obviously, every graph G with on-line chromatic number 4 must contain an induced subgraph G' such that $\chi_{FF}(G') = 4$. In [GKL1] we list all graphs of FF-chromatic number 4 which are minimal. From the list of these 22 graphs the $\boldsymbol{\Delta}$-free ones are F_i, $1 \leq i \leq 5$, shown in Figure 2. It is also shown in [GKL1] that F_2, F_3, and F_4 are 4-critical graphs, $F_1 = \mathbf{B}$ and F_5 are not. This is formulated in the following proposition.

(4.1) Let G be a $\boldsymbol{\Delta}$-free 4-critical graph. Then either G is isomorphic to one of F_2, F_3, and F_4 or G contains at least one of $\mathbf{B}$ and F_5, shown in Figure 2. □

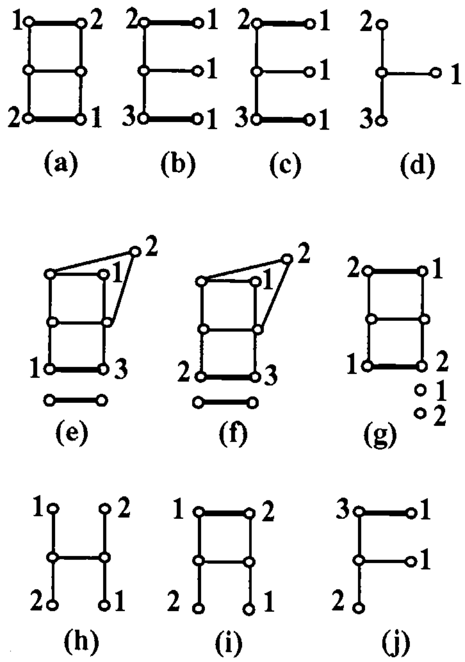


FIG. 5.

The list of all 4-critical graphs containing F_5 is obtained in [GKL1] and shown in Figure 3. The analysis performed in this section results in a list of 4-critical graphs containing a copy of $\mathbf{B}$; see Figure 4. First we show that every graph in Figure 4 has on-line chromatic number 4. Then we prove that the list contains 4-critical graphs and is complete. We discuss connected and disconnected graphs separately in sections 4.1 and 4.2.

To prove $\chi^*(B_i) \geq 4$ we show that Drawer has a 4-forcing strategy against Painter for every $1 \leq i \leq 10$. Let $v_1, v_2, \ldots$ be the order of vertices of G as revealed by Drawer, and let D_k be the colored subgraph after the kth step of the coloring game.

BE-strategy. Let D_4 be isomorphic to $\mathbf{II}$. If D_4 becomes 2-colored, then Drawer wins on $\mathbf{B}$ (see Figure 5(a)). If D_4 is 3-colored, say, $(1,2)$ and $(1,3)$ are the colored edges, then let v_5 be an isolated vertex. Painter essentially has two different choices to color v_5. In both cases Drawer wins on $\mathbf{E}$ (see Figures 5(b) and 5(c)). It is easy to check that $\mathbf{E} \subset B_i$ for $1 \leq i \leq 4$; thus $\chi^*(B_i) \geq 4$ is satisfied by these graphs.

Pigeonhole strategies. Let v_1, v_2, v_3, v_4 be isolated vertices. If D_4 contains three vertices of the same color, say, v_1, v_2, v_3 are colored 1, then Drawer reveals v_5, v_6, v_7 with edges v_1v_5, v_2v_6, and v_3v_7. In D_7 two of these edges have the same coloring pattern, say, $(1,2)$, and Drawer wins on $\mathbf{B}$ (see Figure 5(a)). If D_4 contains three vertices of distinct colors, then Drawer wins on a "claw" (see Figure 5(d)). This strategy is feasible if the graph has $3K_2 + K_1$. Therefore, one may assume that for every B_i, $i = 7, 8$, and 9, D_4 is 2-colored according to the pattern $(1,1,2,2)$. From the fifth step the strategy depends on the graph in question. For B_7, a fifth isolated vertex v_5 is Drawer's winning move. Indeed, by this move Drawer forces three vertices of the same color or three distinctly colored vertices; both are winning positions for Drawer as before. For B_8, the winning position is $4K_2$. In that case D_8 always has two edges

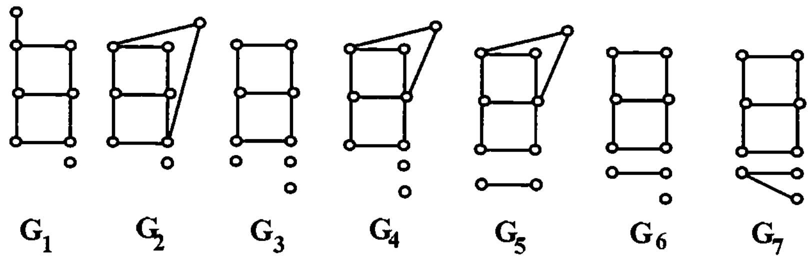


FIG. 6.

with the same coloring pattern, by the pigeonhole principle, and then Drawer wins on **B**.

Ξ-variant. In the case of B_9 and B_{10} Drawer starts with the pigeonhole strategy and obtains three isolated vertices v_1, v_2, v_3 with coloring pattern $(1, 1, 2)$. Then v_4 is given by Drawer with edge v_1v_4 and a new edge, v_5v_6 (so until now a Ξ is given). If Painter uses color 3, then Drawer wins as in Figure 5(e) or as in Figure 5(f); otherwise the edges are colored by 1 and 2 and Drawer wins as in Figure 5(g).

A-variant. The strategy for B_5 and B_6 differs from the fourth step; however, its elements are the same as before. After the first three isolated vertices D_3 contains two vertices with the same color, say, v_1, v_2 are colored 1 and v_3 is colored 2. Then Drawer's winning move consists in giving v_4 with edge v_1v_4. Depending on the color of v_4 (2 or 3) Drawer wins on graph **A** or on **F** (see Figures 5(i) and 5(j)).

In the next step of our analysis we show that all graphs in Figure 4 are 4-critical. The removal of any vertex of B_i, $1 \le i \le 4$, results in a ($\boldsymbol{\Delta}$-free) graph which is either Ξ-free or **B**- and F_5-free. In the first case the proper subgraphs have on-line chromatic number at most 3, by Theorem 1. In the second case FF is obviously a 3-coloring (c.f. (4.1)). Hence B_i, is 4-critical for $1 \le i \le 4$. To see that B_i is 4-critical for $5 \le i \le 10$ it is enough to check the on-line chromatic number of its proper subgraphs containing **B** (otherwise FF is a 3-coloring). Among all of these graphs it is enough to deal with the maximal ones: G_j, $1 \le j \le 7$, listed in Figure 6.

Since algorithm $\mathcal{A}$ defined in section 3 works also for ($\boldsymbol{\Delta}$-free) graphs with a Ξ-free connected component plus any number of isolated vertices, $\chi_{\mathcal{A}}(G_j) \le 3$ follows for $1 \le j \le 4$. The on-line 3-colorability of G_5, G_6, and G_7 will be settled in section 4.2.

4.1. Connected 4-critical graphs. Let G be a connected $\boldsymbol{\Delta}$-free graph containing **B**. The main goal of the present section consists of proving Theorem 2, which states that the following statements are equivalent:

(1) G is **E**-free.

(2) G has no induced subgraph isomorphic to any of F_3, F_4 in Figure 2 and B_1, B_2, B_3, B_4 in Figure 4.

(3) G is Ξ-free.

(4) G has on-line chromatic number $\chi^*(G) \le 3$.

Our algorithm $\mathcal{A}$ in the proof of Theorem 1 is an on-line 3-coloring whenever G is Ξ-free; thus we have (3) $\Longrightarrow$ (4). If G contains both **B** and **E**, then Drawer may use the **BE**-strategy mentioned above and forces a 4-coloring; hence (4) $\Longrightarrow$ (1). Observe that all graphs in (2) contain a copy of **E**, thus (1) $\Longrightarrow$ (2) . Therefore, it is enough to prove the remaining implication (2) $\Longrightarrow$ (3).

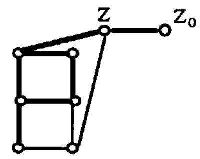


FIG. 7.

(4.2) Let G be a connected $\mathbf{\Delta}$-free graph containing $\mathbf{B}$. Then G is $\mathbf{\Xi}$-free if and only if G has no induced subgraph isomorphic to any of F_3, F_4 in Figure 2 and B_1, B_2, B_3, B_4 in Figure 4.

Proof. Since all forbidden graphs contain $\mathbf{\Xi}$, necessity is obvious. We prove sufficiency by contradiction. Suppose there exists a minimal counterexample G containing $\mathbf{\Xi}$. Let D be a maximal bipartite $\mathbf{\Xi}$-free subgraph of G such that it contains a copy of $\mathbf{B}$. First we show that every vertex of $V(G) \setminus V(D)$ has a neighbor in D. Suppose to the contrary that there are vertices $z, z_0 \notin V(D)$ such that zz_0 is an edge, z_0 has no neighbors in D, and $D + z$ is connected. By the minimality of G and by the choice of D, it follows easily that $D = \mathbf{B}$ and z is an odd extension of $\mathbf{B}$. Consequently, $G = (\mathbf{B} + z) + z_0$ is the graph shown in Figure 7, which contains F_3, a contradiction. Hence $D + z$ is connected for every $z \in V(G) \setminus V(D)$.

The proof of (4.2) (i.e., that the counterexample G does not exist) is arranged in three steps. Let z be called an *illegal extension* of D if $D + z$ contains $\mathbf{\Xi}$. In Steps 1 and 2, we show that D has no illegal (bipartite or odd) extension. In Step 3 we prove that the set of all odd extensions of D satisfy the conditions required by the structure theorems in section 2. The contradiction is obtained by (2.7), which implies that G is $\mathbf{\Xi}$-free.

Step 1. We show that the bigraph $D = [X, Y]$ has no illegal bipartite extension. Suppose on the contrary that $z \in V(G) \setminus V(D)$ is an illegal bipartite extension of D. By symmetry, one may assume that z extends X, which is adjacent to some vertex of Y. Note also that z is nonadjacent to some vertex of Y (since otherwise it would not be illegal). To get a contradiction, we shall show that $D+z$ contains one of F_3, B_1, B_2, and B_3 or, equivalently, the bipartite complement $\widehat{D+z}$ contains one of $\widehat{F_3} = P_6 + K_1$, $\widehat{B_1} = P_5 + 2K_1$, $\widehat{B_2} = F_2$ (see Figure 2), and $\widehat{B_3} = \mathbf{E} + K_2$. For convenience, we are working on the bipartite complement of G, and $G^* = \widehat{D+z}$ is considered as the extension of $\widehat{D}$. Note that Y contains both neighbors and nonneighbors of z also in G^*. Let $G_i = [X_i, Y_i]$ be the nontrivial connected components of $\widehat{D}$, $1 \leq i \leq k$. By (2.4), each G_i is a connected reduced halfgraph. Since D contains $\mathbf{B}$, and since $\widehat{\mathbf{B}} = \mathbf{\Xi}$, we have $k \geq 2$. From the assumption that z is an illegal extension it follows that G^* has a P_5.

Assume that G_1 has a pair of nonadjacent vertices $x \in X_1$, $y \in Y_1$. Supposing that z is (uniformly) nonadjacent to Y_1 any P_5 avoids G_1 and together with $\{x, y\}$ induces a $P_5 + 2K_1 \subset G^*$. Suppose now that z is uniformly adjacent to Y_1, and consider a P_5 induced by $\{x_1, y, z, y_2, x_2\}$, where $x_1 \in X_1$, $x_2 \in X_2$, and $y_2 \in Y_2$ (a P_5 in this form must exist). Then some $y' \notin Y_1$ is nonadjacent to z. Hence $\{x_1, y, z, y_2, x_2, y', x\}$ induces a $P_6 + K_1$ or $P_5 + 2K_1$ in G^*, depending on whether x_2y' is an edge (see Figure 8(a)). As a corollary, one may assume that for each G_i ($1 \leq i \leq k$) different from the complete bigraph, $G_i + z$ contains one of L_1 and L_2 in Figure 8 as an induced subgraph. (z has both neighbors and nonneighbors in Y_i and $[X_i, Y_i]$ is a connected reduced halfgraph.)

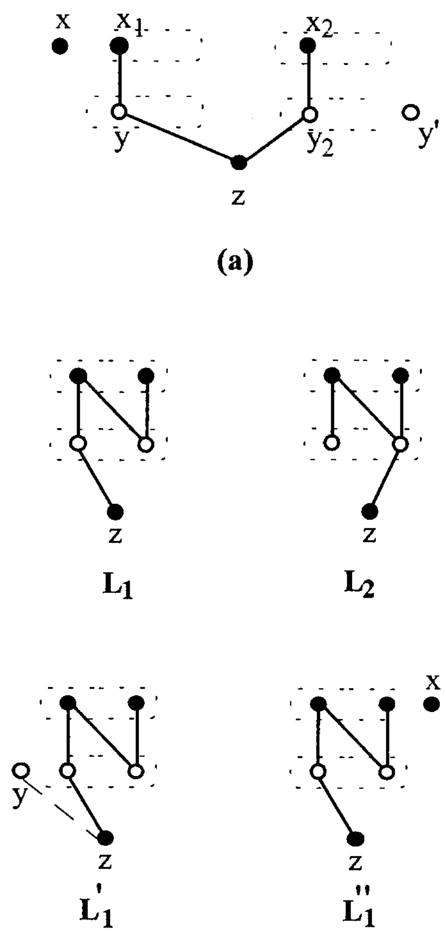


FIG. 8.

If there exist two components different from the complete bigraph, then G^* contains the union of L_i and L_j $(1 \le i \le j \le 2)$ sharing a common vertex z. Since this union contains a $P_6 + P_1$ for each of the three possible choices of (i, j), one may assume that all but possibly one component is a complete bigraph.

Case a. G_1 is not a complete bigraph.

First suppose that $L_1 \subset G_1 + z$. To get a contradiction we show that there exists a copy of L_1 and there are two nonadjacent vertices $x \in X$, $y \in Y$ not in L_1 such that x and y have no neighbor in $L_1 - z$. Since $k \ge 2$ and G_2 contains an edge x_2y_2, the claim follows if G^* has at least one more component. If this is not true, then (since $\mathbf{B} \subset D$) it follows easily that $G_1 + z$ contains either L_1' or L_1'' in Figure 8 with x or y in G_1. Obviously, L_1 and x_2y_2 together with x or y contain either a $P_5 + 2K_1$ or a $P_6 + K_1$.

Suppose now that $L_2 \subset G_1 + z$. If $G_1 + z$ has a P_5 (in this case it must have an L_1 as well), then we are done by using the previous argument. Hence there exists an edge zy, for some y not in G_1. Let $x_2y_2 \in E(G_2)$. If x_2y is a nonedge, then $V(L_2) \cup \{x_2, y_2, y\}$ induces either an $\mathbf{E} + K_2$ or an F_2, depending on whether z and y_2 are nonadjacent or adjacent (see Figure 9(a)). If $y \ne y_2$, zy_2 is a nonedge and x_2y

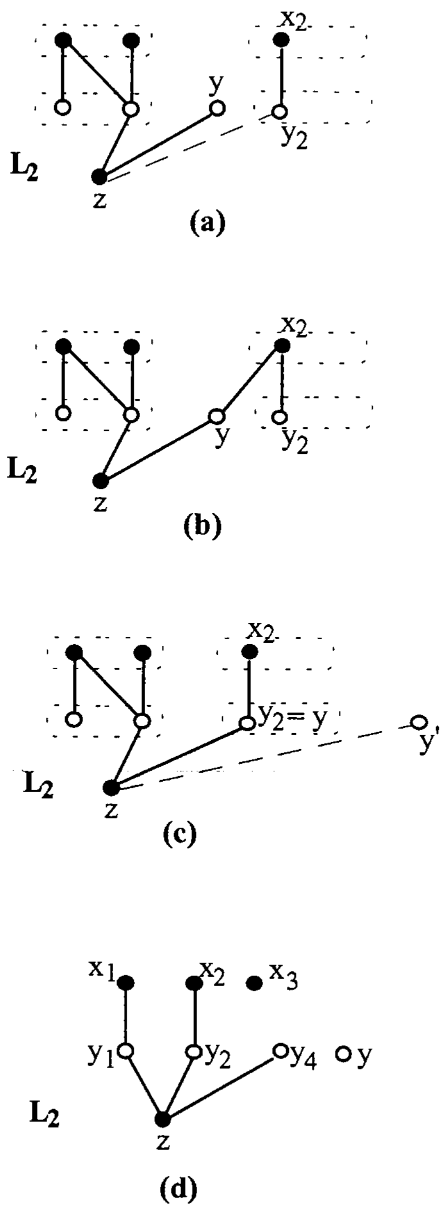


FIG. 9.

is an edge, then the subgraph induced by $V(L_2) \cup \{x_2, y_2, y\}$ contains $P_6 + K_1$ (see Figure 9(b)). Hence one may assume that $y = y_2$. If some $y' \notin Y_1$ is nonadjacent to x_2, then the subgraph induced by $V(L_2) \cup \{x_2, y_2, y'\}$ either contains a $P_6 + K_1$ or induces an F_2 (see Figure 9(c)). If this last condition does not hold, then (using $\mathbf{B} \subset D$) we easily obtain the existence of $y' \in Y_1$ nonadjacent to L_2. Then the subgraph induced by $V(L_2) \cup \{x_2, y_2, y'\}$ contains a $P_6 + K_1$.

Case b. G_i is a complete bigraph for every $i = 1, \ldots, k$.

Suppose that (x_1, y_1, z, y_2, x_2) is a P_5 with $x_i \in X_i$ and $y_i \in Y_i$ $(i = 1, 2)$. Since $\mathbf{B} \subset D$, there are two more components (possibly trivial) containing vertices $x_3 \in X$ and $y_4 \in Y$. One may assume that $zy_4 \in E(G^*)$ (because otherwise a $P_5 + 2K_1$ is found). Then by the connectivity of $D + z$, we have $yz \notin E(G^*)$ for some $y \in Y$. If yx_3 is a nonedge, then we get a $P_5 + 2K_1$; otherwise, $\{x_1, y_1, z, y_2, x_2, y_4, y, x_3\}$ induces an $\mathbf{E} + K_2$ (see Figure 9(d)).

In each case there is a forbidden configuration; therefore, D has no illegal bipartite extension.

Step 2. Next we show that D has no illegal odd extension. Suppose to the contrary that $z \in V(G) \setminus V(D)$ is an illegal odd extension of the bigraph $D = [X, Y]$. Consider the modular decomposition of D and let $M_i = [X_i, Y_i]$, $1 \le i \le k$, be the nontrivial modules. Since D contains a $\mathbf{B}$, we have $k \ge 2$. Denote by $X(z) \subset X$ and $Y(z) \subset Y$ the set of all neighbors of z in X and Y, respectively. Since z is an odd extension of D and G is $\mathbf{\Delta}$-free, $X(z)$ and $Y(z)$ are nonempty sets belonging to the same module, say, M_1, and $X(z) \cup Y(z)$ is an independent set of D. Clearly, $G_X = (D + z) - X(z)$ and $G_Y = (D + z) - Y(z)$ are bipartite proper subgraphs of G. Moreover, one of them contains Ξ (since z is an illegal extension). Hence, by the minimality of G, $\Xi \subset G_X$ (or $\Xi \subset G_Y$) implies that G_X (or G_Y) either is disconnected or $\mathbf{B}$-free. This observation implies that $k = 2$ as follows. If $k \ge 3$, then G_X (and G_Y) is connected, and any $\mathbf{II}$ between M_2 and M_3 together with z would induce Ξ in G_X (and G_Y). Consequently, G_X (and G_Y) must be $\mathbf{B}$-free. In particular, $k \le 3$ and D has no trivial module. Suppose that $k = 3$. If M_1 has an edge xy, then one of its end vertices is not in $X(z) \cup Y(z)$, say, $x \notin X(z)$. Then clearly G_X has a $\mathbf{B}$, which is not allowed. Hence M_1 has no edge. If one of M_2 and M_3 has an edge, then G_X (and G_Y) also contains a $\mathbf{B}$. We have obtained that D has exactly three modules, none of which contains an edge. Then D has no $\mathbf{B}$, a contradiction. Therefore, $k = 2$ follows.

Next we show that $M_1 + z$ is an extended halfgraph. Suppose that $H(n)$ is a primitive representative of M_1 with partite sets $\{x_1, \dots, x_n\}$ and $\{y_1, \dots, y_n\}$ with $x_i y_j$ an edge if and only if $i < j$. It is enough to prove that at least one of $\mathcal{C}_{M_1}(x_t)$ and $\mathcal{C}_{M_1}(y_t)$ is uniformly adjacent to z for every $1 \le t \le n$. First assume that there exist vertices $x \in \mathcal{C}_{M_1}(x_t) \cap X(z)$, $\tilde{x} \in \mathcal{C}_{M_1}(x_t) \setminus X(z)$, and suppose on the contrary that $\tilde{y} \in \mathcal{C}_{M_1}(y_t) \setminus Y(z)$. (By the symmetry of $\mathcal{C}_{M_1}(x_t)$ and $\mathcal{C}_{M_1}(y_t)$, our argument also applies when the roles of X and Y are interchanged.)

Let $x' \in X_2$ and $y' \in Y_2$ be nonadjacent vertices of $D - M_1$. If $t < n$, then let us choose an arbitrary vertex $y'' \in \mathcal{C}_{M_1}(y_n)$. Since $x \in X(z)$ and $xy'' \in E(D)$, we have $y'' \notin Y(z)$. Then the set $\{z, x, \tilde{x}, \tilde{y}, x'', y'', y'\}$ induces an F_3 (see Figure 10(a)), a contradiction. The same contradiction can be deduced for $t = n$ if there exists a vertex $y'' \in Y \setminus Y_1$ adjacent to x'.

We analyze further the case $t = n$ assuming that $D - M_1$ has no Ξ with its isolated vertex in Y. Then, from the condition $\mathbf{B} \subset D$, it follows easily that D contains a copy of $\mathbf{B}$ such that $x'y_i$, $y'x_j$ $(1 \le i \le j \le n)$ are the top and bottom edges (i.e., edges of the $\mathbf{II}$-part) and x^*y^* with $x^* \in X$, $y^* \in Y_1$ is the middle edge (i.e., the edge between the two star vertices). Observe that any vertex of $\mathcal{C}_{M_1}(y_n)$ may play the role of y^*; thus (in the present case) we may set $y^* = \tilde{y}$. If $x_j \in X(z)$, then we get an F_3 (see Figure 10(b)). Thus we assume that $x_j \notin X(z)$ holds. Supposing that $x^* \notin X_1$, we may choose for y_i any vertex of Y_1 adjacent to z. This would result in a copy of F_3 (see Figure 10(c)); thus we may assume that $x^* \in X_1$ holds.

Supposing that $x^* \in X(z)$ (and because $x_j \notin X(z)$), we obtain a B_1 (see Figure 10(d)). Thus we may assume that $x^* \notin X(z)$ also holds. Regardless of whether zy_i

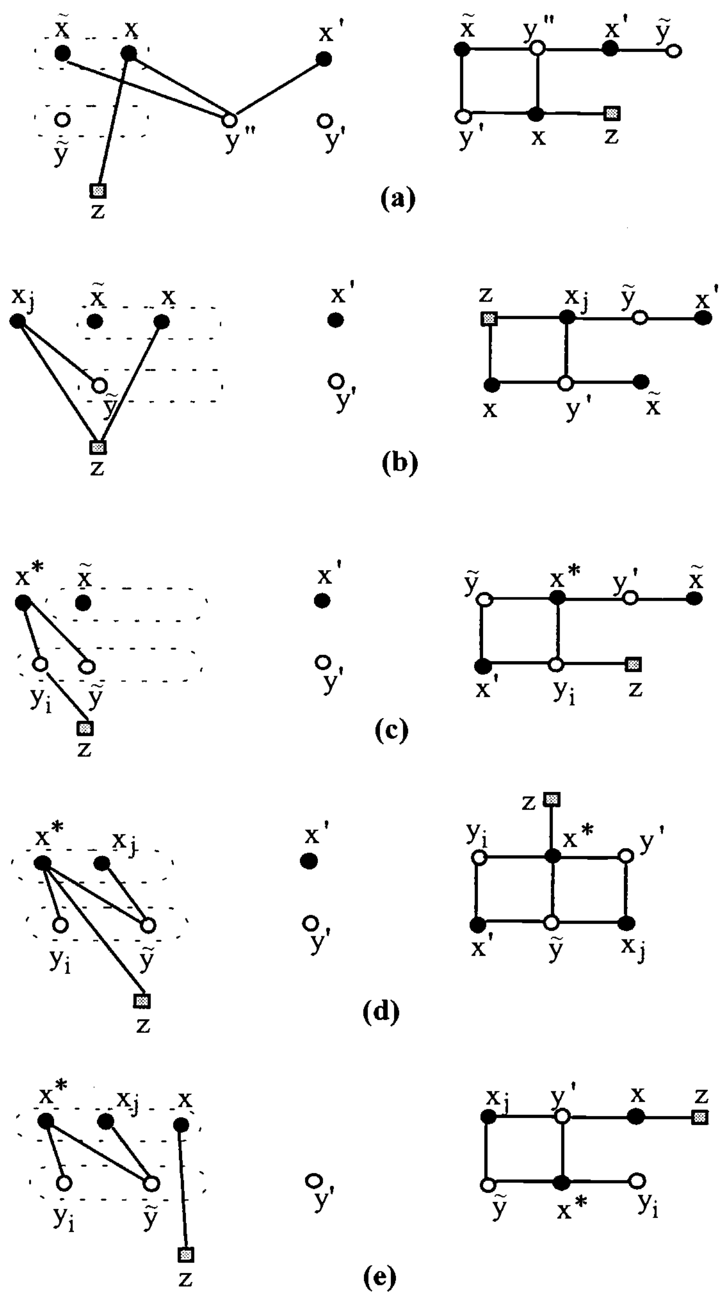


FIG. 10.

is an edge or a nonedge, we get an F_3; see Figure 10(e) if $y_i \notin Y(z)$ and Figure 10(c) otherwise. Thus we have obtained that the existence of the vertices $x \in \mathcal{C}_{M_1}(x_t) \cap X(z)$ and $\tilde{x} \in \mathcal{C}_{M_1}(x_t) \setminus X(z)$ $(1 \leq t \leq n)$ implies that $\mathcal{C}_{M_1}(y_t) \subset Y(z)$. Recall that the same is true when interchanging the role of X and Y.

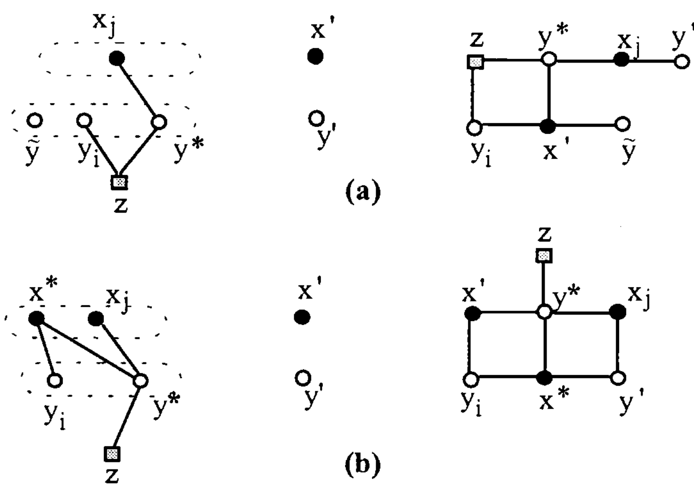


FIG. 11.

In the next step we show that $\mathcal{C}_{M_1}(y_1) \cap Y(z) \neq \emptyset$. Suppose that $\tilde{y} \notin Y(z))$ holds for every $\tilde{y} \in \mathcal{C}_{M_1}(y_1)$. Notice that $Y(z) \neq \emptyset$ implies that $n \geq 2$. Therefore (since G is $\mathbf{\Delta}$-free), there exist vertices $\tilde{x} \in \mathcal{C}_{M_1}(x_1) \setminus X(z)$ and $x \in X(z)$. If there is a vertex $y'' \in Y \setminus Y_1$ adjacent to x', then we get F_3 (see Figure 10(a)). Otherwise, consider again the copy of $\mathbf{B}$ with vertex set $\{x_j, y_i, x^*, y^*, x_j, x', y'\}$, where $y^* \in \mathcal{C}_{M_1}(y_n)$ and $x^* \in \mathcal{C}_{M_1}(x_1) \cup (X \setminus X_1)$. Notice that $x^* \notin X(z)$ holds (by the same argument as before).

If $x_j \in X(z)$, then we get F_3 as in Figure 10(a), with $x = x_j$, $\tilde{x} = x^*$, and $y'' = y^*$. Thus we may assume that $x_j \notin X(z)$ holds for every vertex in the role of x_j. Therefore (since $X(z) \neq \emptyset$), there exists $x \in \mathcal{C}_{M_1}(x_n) \cap X(z)$. If the situation is different from the one in Figure 10(e) (with $\tilde{y} = y^*$), then either $y_i \in Y(z)$ or $y^* \in Y(z)$ holds, but not both, since in this case we get an F_3 (see Figure 11(a)).

In either case we get an F_3: see Figure 10(c) (with $\tilde{y} = y^*$) if $y_i \in Y(z)$, and see Figure 11(b) if $y^* \in Y(z)$.

This proves that $\mathcal{C}_{M_1}(y_1) \cap Y(z) \neq \emptyset$. By the symmetry of halfgraphs, the same argument shows that $\mathcal{C}_{M_1}(x_n) \cap X(z) \neq \emptyset$. From the previous steps of the proof it follows that at least one of the properties $\mathcal{C}_{M_1}(y_t) \subset Y(z)$ and $\mathcal{C}_{M_1}(x_t) \subset X(z)$ holds for $t = 1$ and n.

To conclude the proof, suppose that there exist vertices $\tilde{x} \in \mathcal{C}_{M_1}(x_t) \setminus X(z)$ and $\tilde{y} \in \mathcal{C}_{M_1}(y_t) \setminus Y(z)$, for some $1 < t < n$. Let $y^* \in \mathcal{C}_{M_1}(y_n)$ and $x^* \in \mathcal{C}_{M_1}(x_1)$. If $y^* \in Y(z)$, then we get F_3 as in Figure 11(b) (with $x_j = \tilde{x}$ and $y_i = \tilde{y}$). If $x^* \in X(z)$, then we get F_3 as in Figure 10(c) (with $x_j = \tilde{x}$, $y_i = \tilde{y}$ and $\tilde{y} = y^*$). Assuming that $x^* \notin X(z)$, $y^* \notin Y(z)$, and choosing a vertex $x \in \mathcal{C}_{M_1}(x_n) \cap X(z)$, we get F_3 as in Figure 10(e) (with $x_j = \tilde{x}$, $y_i = \tilde{y}$, and $\tilde{y} = y^*$).

Hence, for every $1 \leq t \leq n$, at least one of $\mathcal{C}_{M_1}(x_t)$ and $\mathcal{C}_{M_1}(y_t)$ is uniformly adjacent to z. This implies that $M_1 + z$ is an extended halfgraph. In particular, by (2.7), $G + z$ is $\mathbf{\Xi}$-free, a contradiction. Therefore, D has no illegal odd extension.

Step 3. If the subgraph $G - z$ is bipartite for some $z \in V(G)$ and contains $\mathbf{B}$, then by the choice of D, $D = G - z$; furthermore, z is an illegal extension of D. This is not possible as we have seen in Steps 1 and 2. Therefore, by the maximality of D, one may assume that $G - D$ has at least two vertices, and each $z \in V(G - D)$ is a

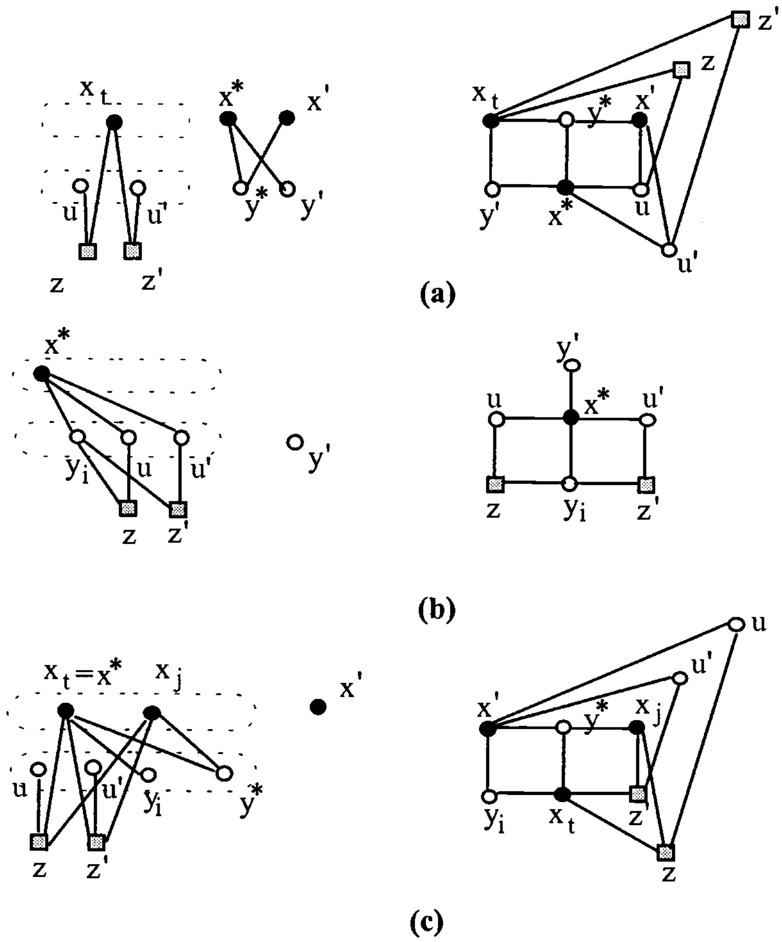


FIG. 12.

legal odd extension of D. Hence, by the structure result in section 2, D has exactly two nontrivial modules.

Let Z_i be the set of all odd extensions of D adjacent to M_i, $i = 1, 2$. By (2.6), for every $z \in Z_i$, $M_i + z$ form an extended halfgraph. By Step 2, and since G is $\mathbf{\Delta}$-free, Z_1 and Z_2 are independent sets.

First we prove that, for $Z_1 \neq \emptyset$, the bigraph $H = [Z_1, Y_1]$ is $\mathbf{II}$-free. Then, by symmetry, the same is true for the bigraphs $[Z_i, Y_i]$, $[Z_i, X_i]$ $(i = 1, 2)$. Let $\{x_1, \ldots, x_n\}$ and $\{y_1, \ldots, y_n\}$ be the partite sets of a primitive representative $H(n)$ of M_1 with $x_i y_j \in E(D)$ if and only if $i < j$. For any $z \in Z_1$, let $X(z) = \{x \in X_1 \ : \ xz \in E(G)\}$ and $Y(z) = \{y \in Y_1 \ : \ yz \in E(G)\}$. Assume that zu and $z'u'$ are edges of a $\mathbf{II} \subset H$, where $z, z' \in Z_1$ and $u, u' \in Y_1$. If u and u' are not equivalent in M_1, then there exists a vertex $x \in X_1$ such that, say, ux is an edge but $u'x$ is not. But if zx is an edge, then zux is a triangle; if it is not, then $zu'x$ is an empty triangle. Thus u and u' belong to the same equivalence class, $\mathcal{C}_{M_1}(y_t)$, for some $1 \leq t \leq n$.

Let $x' \in X \setminus X_1$ and $y' \in X \setminus X_1$ be nonadjacent vertices of $D - M_1$ and choose a copy of $\mathbf{B}$ such that $x'y_i$ and $y'x_j$ $(1 \leq i \leq j \leq n)$ are the top and bottom edges and

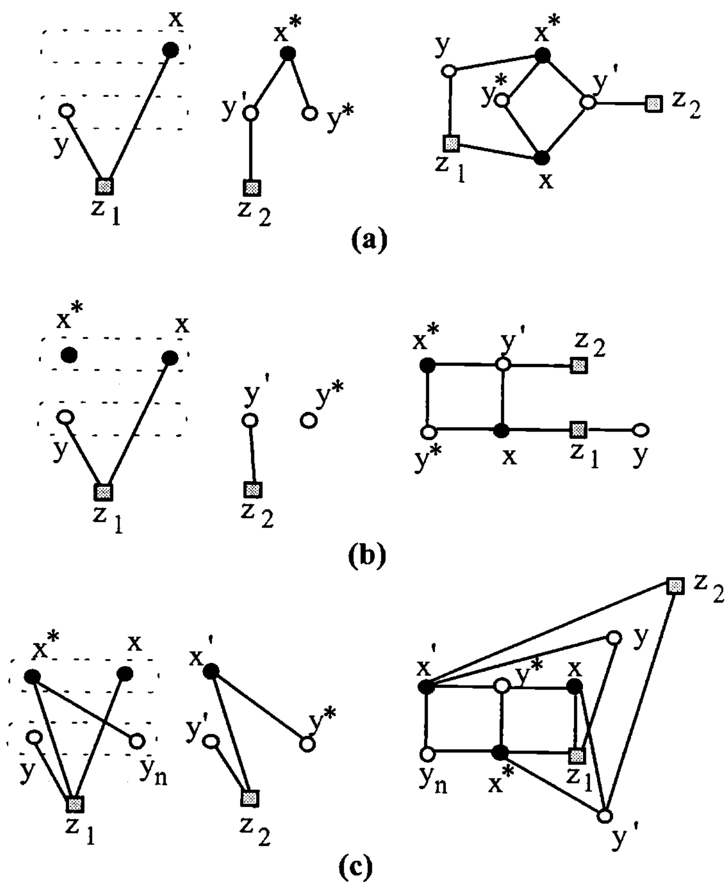


FIG. 13.

x^*y^* with $x^* \in X$, $y^* \in Y$ is the middle edge. If $x^* \in X \setminus X_1$ and $y^* \in Y \setminus Y_1$, then (since $x_t \in X(z) \cap X(z')$) we get a copy of B_4 (see Figure 12(a)).

One may assume that at least one of x^* and y^* is in M_1. This implies that $n \geq 2$; furthermore, every vertex of $\mathcal{C}_{M_1}(y_n)$ or $\mathcal{C}_{M_1}(x_1)$ may play the role of $x^* \in X_1$ or $y^* \in Y_1$. If $1 < t < n$, we obtain a B_4 (see Figure 12(a) with $x^* = x_1$, $y^* = y_n$). If $t = n$, then either $y^* \notin Y_1$ or y_i must be different from $y^*, u, u' \in \mathcal{C}_{M_1}(y_n)$. Choosing $x^* = x_1$, we get B_4 as in Figure 12(a) or we get B_1 (see Figure 12(b)).

If $t = 1$, then either $x^* \notin X_1$ or y_i must be different from $u, u' \in \mathcal{C}_{M_1}(y_1)$. Choosing $y^* = y_n$, we get B_4 as in Figure 12(a), and for the second case, see Figure 12(c). This proves that $[Z_1, Y_1]$ is **II**-free, and, by symmetry, the same is true for each $[Z_i, Y_i]$, $[Z_i, X_i]$ $(i = 1, 2)$.

Let $z_i \in V(G - D)$ $(i = 1, 2)$ be two odd extensions of D adjacent to module M_i of D. We prove that $z_1z_2 \in E(G)$. Suppose on the contrary that this is not true. For $i = 1, 2$, let $H(n_i)$ be the primitive representative of M_i, and assume that $n = n_1 \geq n_2$. Let $\{x_1, \ldots, x_n\}$ and $\{y_1, \ldots, y_n\}$ be the partite sets of $H(n)$ with $x_iy_j \in E(D)$ if and only if $i < j$. Let $x \in X(z_1) \cap \mathcal{C}_{M_1}(x_n)$, $y \in Y(z_1) \cap \mathcal{C}_{M_1}(y_1)$, and $y' \in Y(z_2)$. If M_1 has no edge (that is, $n = 1$), then neither has M_2 (recall that $n \geq n_2$). Then $\mathbf{B} \subset D$ implies that $(D - M_1) - M_2$ contains an edge x^*y^*. Thus we

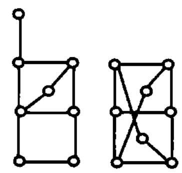


FIG. 14.

get an F_4 (see Figure 13(a)). From now on $n \geq 2$.

Suppose that $D - M_1$ has a vertex nonadjacent with z_2, say, $y^* \notin Y(z_2)$. Let us choose vertices $x^* \in \mathcal{C}_{M_1}(x_1)$ and $x' \in X(z_2)$ adjacent with y^*. (Notice that x' can be chosen from M_2, by Step 2.) Then we get one of F_3 and B_4; see Figure 13(b) if $x^* \notin X(z_1)$ and see Figure 13(c) if $x^* \in X(z_1)$. Thus we may assume that $X(z_2) \cup Y(z_2) = X_2 \cup Y_2$.

Consider the copy of $\mathbf{B}$ defined above, and observe that (in the present case) both x^* and y^* are in M_1. Hence, we may assume that $x^* \in \mathcal{C}_{M_1}(x_1)$ and $y^* \in \mathcal{C}_{M_1}(y_n)$. It follows from $1 < i \leq j < n$ that $x_{j'} \in X(z_1)$ and $y_i \in Y(z_1)$ may be assumed. Since G is $\mathbf{\Delta}$-free, one of x^* and y^* is not adjacent to z, say, $y^* \notin Y(z_1)$. Then, by letting $x = x_j$, we obtain one of F_3 and B_4; see Figure 13(b) if $x^* \notin X(z_1)$ and see Figure 13(c) (with y_i in the role of y_n) if $x^* \in X(z_1)$.

In each case there is a forbidden configuration; therefore, $z_1 z_2 \in E(G)$ follows.

To conclude the proof of (4.2) we refer to the structure theorems in section 2. As we have shown in Steps 1–3, the conditions of (2.7) are satisfied by G; therefore, G has no $\mathbf{\Xi}$, a contradiction. ∎

It is worth noting that our list of forbidden graphs in (4.2) is minimal. Obviously, B_1, B_2, B_3, and B_4 must be on the list; hence each contains $\mathbf{B}$. To see this for F_3 and F_4, in Figure 14 we give connected $\mathbf{\Delta}$-free graphs containing $\mathbf{B}$ such that their only subgraph from the list is F_3 and F_4, respectively.

4.2. Disconnected 4-critical graphs. Let G be a disconnected $\mathbf{\Delta}$-free graph containing $\mathbf{B}$. The connected component $G_0 \subset G$ which contains $\mathbf{B}$ is called the *major component* of G. If G is on-line 3-colorable, then the major component must be $\mathbf{\Xi}$-free by the results in section 4.1. Furthermore, $G - G_0$ is $(K_2 + 2K_1)$- and $\mathbf{II}$-free, since B_7 and B_8 in Figure 4 are not 3-colorable. If $G - G_0$ has no edge, then the algorithm $\mathcal{A}$ in the proof of Theorem 1 is obviously 3-color G.

Therefore, when looking for further 4-critical graphs, one may assume that $G - G_0$ has just one component with an edge, called the *secondary component* of G. Moreover, the secondary component is either a C_5 or a $K_{m,n} - K_2$ or an induced subgraph of $K_{m,n} + K_1$ (i.e., in this last case the secondary component is a complete bipartite graph and G possibly has one more isolated vertex).

(4.3) Let G be a disconnected $\mathbf{\Delta}$-free graph containing $\mathbf{B}$. Then G has on-line chromatic number at most 3 if and only if the major component of G is $\mathbf{\Xi}$-free and G has no induced subgraph isomorphic to any of the graphs B_i, $5 \leq i \leq 10$, in Figure 4.

Proof. All excluded graphs are on-line 4-chromatic (see the beginning of this section); thus we have only to prove that the list is complete. We may assume that G has a major component, a secondary component, and possibly one more isolated vertex. Since B_5 and B_6 are not 3-colorable, we may also assume that the major component G_0 has no $H_1 \subset B_5$ and no $H_2 \subset B_6$ (see Figure 15).

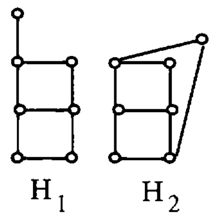


FIG. 15.

First we show that the major component $G_0 \subset G$ is bipartite; moreover, its modules have no edge. To see this, let us consider a maximal bipartite induced subgraph $D \subseteq G_0$ containing $\mathbf{B}$. Clearly, D has at least two nontrivial modules. Let M_1 be a module of D with primitive representative $H(n)$ such that n is as large as possible. Observe that the bipartite complement of H_1 (see Figure 15) is $P_4+K_2+K_1$, and $H(n)$ contains the bipartite complement of P_4+K_1 for $n \geq 3$. Thus we obtain an H_1 in D if $n \geq 3$. Suppose that $n = 2$, and notice that $H(2) = \widehat{P_4}$. It is easy to check that $D \supseteq \mathbf{B}$ implies that the bipartite complement of $D - M_1$ contains K_2+K_1. Thus we obtain an H_1 in D also for $n = 2$. Therefore, $n = 1$ follows; that is, the modules of D have no edge. Since any odd extension of any module of D contains an H_2 (see Figure 15), D obviously has no odd extension. Hence $G_0 = D$ follows and concludes the proof of the claim.

Next we define the required on-line 3-colorings depending on the type of the secondary component. Let $z_1, z_2, \dots$ be the order of vertices of G as revealed by Drawer, and let D_k be the colored subgraph after the kth step of the coloring game. For any integer r, let $A(r)$ denote the set of all vertices of G colored with r by an on-line algorithm A.

Case 1. $G - G_0 = K_2$. To make the definition on the algorithm easier we introduce two new on-line coloring rules. The *equivalence rule* is as follows: if there are some equivalent vertices with the current vertex z, assign to z the minimum color appearing on a z-equivalent vertex. The *parity first fit rule* (PFF) says that the current vertex should be colored by the smallest color which does not appear on a vertex that is at an odd distance from the current one. We define an algorithm $\mathcal{A}^*$ as follows:

- If z_{k+1} is an isolated vertex, D_k has exactly two components but none of them has three different colors, then color z_{k+1} by 2.
- Otherwise, use the equivalence rule when it applies.
- Otherwise, if the component of z_{k+1} in D_{k+1} is not a complete bipartite graph, then apply the PFF rule.
- In any other cases apply the FF rule with two exceptions:
  - If every neighbor of z_{k+1} is colored by 1, there exist a component in D_k which is 2-colored by 1 and 2, there are no 1- and 3-colored vertices in the same component, and there are no both 1- and 2-colored isolated vertices, then color z_{k+1} by 3.
  - If every neighbor of z_{k+1} is colored by 2, there are no 2- and 3-colored vertices in the same component of D_k and there are no both 1- and 2-colored isolated vertices, then color z_{k+1} by 3.

We have to show that $\mathcal{A}^*$ is a 3-coloring for this class. Let G be a graph with major component G_0 and secondary component K_2 such that the modules of the bipartite graph G_0 have no edges. Let $z_1, z_2, \dots$ be an ordering of the vertices of G,

l is a natural number, and $G_0^l = [X, Y]$ is a subgraph of G_0 induced by $\{z_1, \dots z_l\}$. It is not too hard to see that when applying $\mathcal{A}^*$ on G with order $z_1, z_2, \dots$ then after coloring l vertices either X or Y lacks for either any 1-colored or any 2-colored vertices. A similar (but easier) argument shows that the case is similar with 1 and 3 or with 2 and 3. Using these one can check that neither X nor Y can have three different colors. Consequently, there are colors $\{a, b\} \subset \{1, 2, 3\}, a \neq b$, such that X does not have a-colored and Y does not have b-colored vertices. It is obvious that in this case $\mathcal{A}^*$ cannot use more than three colors.

Case 2. $G - G_0 \neq K_2$. In this case $G - G_0$ is a subgraph of either a $K_{m,n} + K_1$ or a $K_{m,n} - K_2$ or a C_5 and either $K_2 + K_1$ or $K_{1,2}$ is contained in it. As B_9 and B_{10} are not contained in G it is easy to see that the edgeless nontrivial modules of G_0 consist of two vertices such that G_0 is a complete bipartite graph with some nonincident edges deleted. We define algorithm $\mathcal{A}^{**}$, which is similar to (but simpler than) $\mathcal{A}^*$, as follows:

- If z_{k+1} is an isolated vertex and D_k has exactly two or exactly three components, then color z_{k+1} by 2.
- Otherwise, use the equivalence rule when it applies.
- Otherwise, if the component of z_{k+1} in D_{k+1} is not a complete bipartite graph, then apply the PFF rule.
- In any other cases apply the FF rule with two exceptions:
  - If every neighbor of z_{k+1} is colored by 1 and there are no 1- and 3-colored vertices in the same component of D_k, then color z_{k+1} by 3.
  - If every neighbor of z_{k+1} is colored by 2 and there are no 2- and 3-colored vertices in the same component of D_k, then color z_{k+1} by 3.

A similar argument as in Case 1 shows that $\mathcal{A}^{**}$ is really a 3-coloring for this class. □

Let us note that by using the algorithms in the proof of (4.3) we obtain that each graph G_j, $j = 5, 6, 7$, in Figure 6 is on-line 3-colorable. Indeed, $\chi_{\mathcal{A}^*}(G_5) \leq 3$, $\chi_{\mathcal{A}^{**}}(G_6) \leq 3$, and $\chi_{\mathcal{A}^{**}}(G_7) \leq 3$ follow.

As a corollary of (4.2) and (4.3) we obtain the list of all 4-critical graphs excluded from on-line 3-chromatic graphs containing $\mathbf{B}$: F_3, F_4 in Figure 3, and B_i, $1 \leq i \leq 10$ in Figure 4. This concludes the proof of Theorem 3.

REFERENCES

[G] A. Gyárfás, *On Ramsey Covering Numbers*, Infinite and Finite Sets, Colloq. Math. Soc. János Bolyai 10, North-Holland/American Elsevier, New York, 1975, pp. 801–816.

[GKL1] A. Gyárfás, Z. Király, and J. Lehel, *On-line* 3*-chromatic graphs.* II. *Critical graphs*, Discrete Math., 177 (1997), pp. 99–122.

[GKL2] A. Gyárfás, Z. Király, and J. Lehel, *On-line graph coloring and finite basis problems*, Combinatorics, Paul Erdős is Eighty, Vol. 1, Bolyai Soc. Math. Stud., János Bolyai Mathematical Society, Budapest, Hungary, 1993, pp. 207–214.

[GKL3] A. Gyárfás, Z. Király, and J. Lehel, *On-line competitive on-line algorithms*, Technical Report TR-9703-1, Department of Mathematics, Eötvös Loránd University, Budapest, Hungary; also available online from http://www.cs.elte.hu/tr97/.

[GL1] A. Gyárfás and J. Lehel, *On-line and first fit colorings of graphs*, J. Graph Theory, 12 (1988), pp. 217–227.

[GL2] A. Gyárfás and J. Lehel, *First fit and on-line chromatic number of families of graphs*, Ars Combin., 29C (1990), pp. 168–176.

[GL3] A. Gyárfás and J. Lehel, *Effective on-line coloring of* P_5*–free graphs*, Combinatorica, 11 (1991), pp. 181–184.

[I] S. Irani, *Coloring Inductive Graphs On-line*, in Proceedings of the 31st Annual Symposium on the Foundations of Computer Science, 1990, pp. 470–479.

[K] H. A. Kierstead, *The linearity of first-fit coloring of interval graphs*, SIAM J. Discrete Math., 1 (1988), pp. 526–530.

[K1] H. A. Kierstead, *Recursive and On-Line Graph Coloring*, manuscript,

[K2] H. A. Kierstead, *On-Line Coloring k-Colorable Graphs*, manuscript, 1996.

[KK] H. A. Kierstead and K. Kolossa, *On-line coloring of perfect graphs*, Combinatorica, 16 (1996), pp. 479–491.

[KO] K. Kolossa, *On the on-line chromatic number of the family of on-line* 3*-chromatic graphs*, Discrete Math., 150 (1996), pp. 205–230.

[KPT] H. A. Kierstead, S. G. Penrice, and W. T. Trotter, *On-line and first-fit coloring of graphs that do not induce P_5*, SIAM J. Discrete Math., 8 (1995), pp. 485–498.

[KPT1] H. A. Kierstead, S. G. Penrice, and W. T. Trotter, *On-line coloring and recursive graph theory*, SIAM J. Discrete Math., 7 (1994), pp. 72–89.

[KT1] H. A. Kierstead and W. T. Trotter, *An extremal problem in recursive combinatorics*, Congr. Numer., 33 (1981), pp. 143–153.

[KT] H. A. Kierstead and W. T. Trotter, *On-line graph coloring*, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 7, Amer. Math. Soc., Providence, RI, 1992, pp. 85–92.

[L] L. Lovász, *Coverings and colorings of hypergraphs*, Fourth Southeastern Conference on Combinatorics, Graph Theory, and Computing, Utilitas Math., 1973, pp. 3–12.

[LST] L. Lovász, M. Saks, and W. T. Trotter, *An on-line graph coloring algorithm with sublinear performance ratio*, Discrete Math., 75 (1989), pp. 319–325.

[S] D. P. Sumner, *Subtrees of a graph and the chromatic number*, in Theory and Applications of Graphs, G. Chartrand, ed., John Wiley, New York, 1981, pp. 557–576.

[V] S. Vishwanathan, *Randomized online graph coloring*, J. Algorithms, 13 (1992), pp. 657–669.

SIAM J. DISCRETE MATH.
Vol. 12, No. 3, pp. 412–412

NOTE: AN UPPER BOUND FOR THE DIAMETER OF A GRAPH*

RUSSELL MERRIS†

Abstract. In an article [*SIAM J. Discrete Math.*, 7 (1994), pp. 443–457], Chung, Faber, and Manteuffel established an inequality for the diameter of a graph in terms of its Laplacian eigenvalues, one that could be stated more attractively if graphs with a given property did not exist. The purpose of this note is to exhibit graphs having that given property.

Key words. Laplacian, diameter, eigenvalues

AMS subject classification. 05C50

PII. S0895480195282471

Let $G = (V, E)$ be a graph with vertex set $V = \{1, 2, \ldots, n\}$ and edge set $E \subset V^{(2)}$, the 2-element subsets of V. The Laplacian matrix, $L(G) = (q_{ij})$, is defined by q_{ij} = the degree of vertex i when $j = i$, -1 when $j \neq i$, but $\{i, j\} \in E$, and 0 otherwise. It is well known that $L(G)$ is positive semidefinite and singular. Denote its eigenvalues by $0 \leq \lambda_2 \leq \cdots \leq \lambda_n$. In [1, Thm. 5.3], an upper bound for the diameter, $D(G)$, is obtained in terms of λ_2 and λ_n, provided $G \neq K_n = (V, V^{(2)})$. In a remark on p. 448 of [1], the authors observe that their result "would be more aesthetically pleasing if it yielded

$$D(G) \leq \left\lceil \frac{\cosh^{-1}(n-1)}{\cosh^{-1}\left(\dfrac{\lambda_n + \lambda_2}{\lambda_n - \lambda_2}\right)} \right\rceil,$$

[where $\lceil\ \ \rceil$ is the ceiling function]. This inequality fails only if $D(G) = m + 1$ with

$$m = \frac{\cosh^{-1}(n-1)}{\cosh^{-1}\left(\dfrac{\lambda_n + \lambda_2}{\lambda_n - \lambda_2}\right)}. \tag{1}$$

Does such a graph exist?"

The answer to the question is yes. Let M be an r-matching of K_n. (Then M is a set of $r \geq 1$ nonadjacent edges of K_n.) If $G = K_n - M = (V, V^{(2)} \backslash M)$, then $D(G) = 2$, $m = 1$, $\lambda_n = n$, and $\lambda_2 = n - 2$, so (1) is satisfied. If M is a perfect matching, then $G = K_n - M$ is the so-called hyperoctahedral or cocktail party graph.

REFERENCE

[1] F. R. K. CHUNG, V. FABER, AND T. A. MANTEUFFEL, *An upper bound on the diameter of a graph from eigenvalues associated with its Laplacian*, SIAM J. Discrete Math., 7 (1994), pp. 443–457.

*Received by the editors March 6, 1995; accepted for publication March 5, 1999; published electronically September 7, 1999.
http://www.siam.org/journals/sidma/12-3/28247.html
†Department of Mathematics and Computer Science, California State University, Hayward, CA 94542 (rmerris@gauss.sci.csuhayward.edu).

SIAM J. DISCRETE MATH.
Vol. 12, No. 4, pp. 413-424

COMPLEXES OF DIRECTED GRAPHS*

ANDERS BJÖRNER† AND VOLKMAR WELKER‡

Abstract. Let P be a monotone property of directed graphs on n vertices, and let Δ_n^P denote the abstract simplicial complex whose simplices are the edge sets of graphs having property P. We prove the following:

1. If "$P =$ acyclic," then Δ_n^P is homotopy equivalent to the $(n-2)$-sphere.
2. If "$P =$ not strongly connected," then Δ_n^P has the homotopy type of a wedge of $(n-1)!$ spheres of dimension $2n-4$.

The lattice of all posets on $\{1, 2, \ldots, n\}$ plays an important role in the analysis. We also discuss some other properties of directed graphs from this point of view.

Key words. directed graph, acyclic, strongly connected, complex of graphs, lattice of posets

AMS subject classifications. 05E25, 05C20, 05C40

PII. S0895480198338724

1. Introduction.

A property of graphs is called *monotone* if it is preserved under the deletion of edges. Thus, a monotone graph property can be interpreted as a simplicial complex, and one can study its topological properties. This has been done for undirected graphs in a number of recent papers. See [1, 13] and the further references cited there.

Here we look at some monotone properties of directed graphs from this point of view. Let $[n] = \{1, 2, \ldots, n\}$. We identify a digraph G on the node set $[n]$ with the set $E(G)$ of its edges, which is a subset of the set $\Omega = [n] \times [n] \setminus \{(i,i) \mid 1 \leq i \leq n\}$. In particular, in this paper digraphs have no multiple edges and no loops.

A digraph is said to be *acyclic* if it contains no directed cycle of edges. Our first result is the following theorem.

THEOREM 1.1. *The complex Δ_n^{ACY} of acyclic digraphs on n vertices is homotopy equivalent to the $(n-2)$-dimensional sphere.*

A digraph is *strongly connected* if for every pair $(i,j) \in \Omega$ there is a sequence (v_k, v_{k+1}), $k = 0, \ldots, f$ of edges in $E(G)$ such that $v_0 = i$ and $v_{f+1} = j$. Thus, the property of being *not* strongly connected is monotone.

THEOREM 1.2. *The complex Δ_n^{NSC} of not strongly connected digraphs on n vertices is homotopy equivalent to a wedge of $(n-1)!$ spheres of dimension $2n-4$.*

The proofs of both theorems, to be given in sections 2 and 3, resp., rely on an analysis of certain properties of the lattice of all partially ordered sets (posets) on the ground set $[n]$. For two such posets P and P', say that P is less than P' if $i < j$ in P implies that $i < j$ in P' for all $(i,j) \in \Omega$. This partial order, augmented by a top element, is a lattice whose proper part we denote by Pos_n. We show that Pos_n

*Received by the editors May 13, 1998; accepted for publication (in revised form) March 11, 1999; published electronically October 19, 1999.

http://www.siam.org/journals/sidma/12-4/33872.html

†Matematiska Institutionen, Kungl. Tekniska Högskolan, S-100 44 Stockholm, Sweden (bjorner@math.kth.se). The research of this author was supported by the Göran Gustafsson Foundation for Research in Natural Sciences and Medicine.

‡Fachbereich Mathematik und Informatik, Philipps-Universität Marburg, 35032 Marburg, Germany (welker@mathematik.uni-marburg.de). The research of this author was supported by Deutsche Forschungsgemeinschaft (DFG).

is homotopy equivalent to the $(n-2)$-sphere by relating it to the covering of the $(n-2)$-sphere given by open hemispheres of the braid arrangement.

In sections 4–6 we comment on complexes of some other classes of digraphs, viz. directed matchings, nonspanning digraphs, and arborescences. We also present some computations and conjectures and remark on the action of the symmetric group on the homology of the graph properties considered in Theorems 1.1 and 1.2.

The computer calculations presented in sections 4 and 6 were done using a program written by Frank Heckenbach.

2. Acyclic digraphs and the lattice of all posets. We will use a number of tools that have become fairly standard in topological combinatorics. For convenience these are summarized in an appendix (section 7), to which we also refer for explanation of notation and terminology.

For any digraph G on the node set $[n]$, let $\widetilde{G}$ denote its *transitive closure.* Thus, (i,j) is an edge of $\widetilde{G}$ if and only if there is a directed path in G from i to j. The mapping $G \to \widetilde{G}$ is a closure operator on the Boolean lattice of all subsets of Ω.

If G is acyclic, then $\widetilde{G}$ is the comparability graph of a poset on $[n]$, and conversely the comparability graphs of posets are precisely the digraphs of the form $\widetilde{G}$ for acyclic digraphs G. Thus the mapping $G \to \widetilde{G}$ restricts to a closure operator on $\overline{L}(\Delta_n^{ACY})$ with image Pos_n.

LEMMA 2.1. Δ_n^{ACY} *and* $\Delta(\mathsf{Pos}_n)$ *are homotopy equivalent.*

Proof. Since $\Delta_n^{ACY} \cong \Delta(\overline{L}(\Delta_n^{ACY}))$ (barycentric subdivision) this follows directly from the closure lemma, Corollary 7.2. ∎

The poset Pos_n consists of all partially ordered sets on the set $[n]$ except for the antichain. As was mentioned, Pos_n is the proper part of a lattice that we denote $\mathsf{Pos}_n^{\#}$. Indeed, in $\mathsf{Pos}_n^{\#}$ the meet of two posets $(P, \leq_P)$ and $(Q, \leq_Q)$ is given by the poset $(R, \leq_R)$ defined by $x \leq_R y$ if and only if $x \leq_P y$ and $x \leq_Q y$. The lattice $\mathsf{Pos}_n^{\#}$ is graded with rank function $\mathrm{rank}(P)$ being the number of strict order relations in P, i.e., the number of P's edges as an acyclic digraph. The bottom element of $\mathsf{Pos}_n^{\#}$ is the antichain (i.e., the poset with empty order relation). The atoms of $\mathsf{Pos}_n^{\#}$ are the posets with exactly one comparability relation, and the coatoms are the $n!$ total orders on $[n]$. In particular, the length of $\mathsf{Pos}_n^{\#}$ is $\binom{n}{2}+1$. Moreover, every poset $P \in \mathsf{Pos}_n^{\#}$ is the join of the atoms below it (whose number equals $\mathrm{rank}(P)$) and the meet of the coatoms above it. The latter is due to the well-known fact that P is the intersection of its linear extensions (these are precisely the coatoms above P). Figure 2.1 shows the poset of posets for $n = 3$.

Theorem 1.1 will follow from Lemma 2.1 and the following.

THEOREM 2.2. $\Delta(\mathsf{Pos}_n)$ *is homotopy equivalent to the* $(n-2)$*-dimensional sphere.*

Proof. Let $X_n = \{\underline{\mathbf{x}} \in \mathbb{R}^n \mid x_1 + \cdots + x_n = 0 \text{ and } \|\underline{\mathbf{x}}\| = 1\}$. This is the unit sphere in an $(n-1)$-dimensional subspace of $\mathbb{R}^n$; hence, X_n is an $(n-2)$-dimensional sphere.

With each atom $i < j$ of Pos_n we associate the open hemisphere $H_{i,j} = \{\underline{\mathbf{x}} \in X_n \mid x_i < x_j\}$. Let N denote the nerve of the covering of X_n by these hemispheres. Thus, N is the simplicial complex whose vertex set is the set A of atoms of Pos_n and whose simplices are the subsets of atoms corresponding to collections of hemispheres with nonempty intersection. By the nerve lemma (see, e.g., [3, Thm. 10.7]) $N \simeq X_n = S^{n-2}$.

It is easy to see that a collection of hemispheres $H_{i,j}$ has nonempty intersection if and only if the corresponding edges $i \to j$ determine an acyclic digraph. The

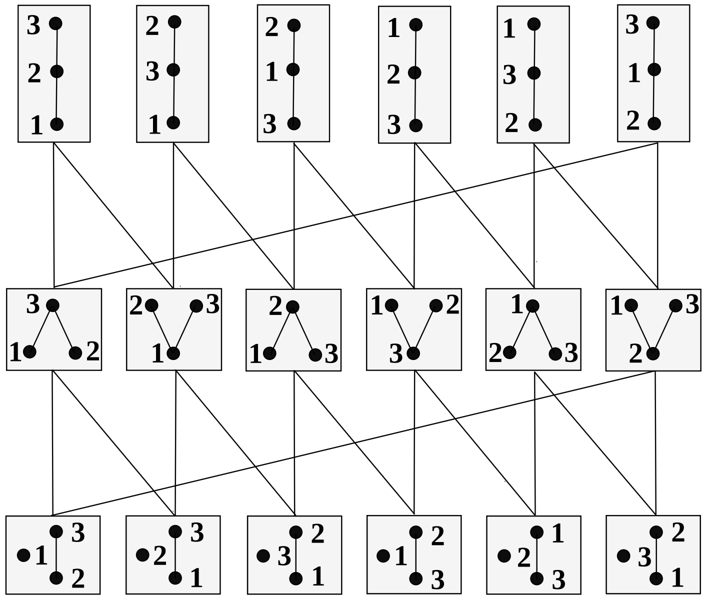


FIG. 2.1. Pos_3.

transitive closure of this acyclic digraph is a poset. Therefore the nerve N equals the crosscut complex $\Gamma(\mathsf{Pos}_n^\#, A)$, and using the crosscut theorem, Proposition 7.4, we get $\Delta(\mathsf{Pos}_n) \simeq \Gamma(\mathsf{Pos}_n^\#, A) = N \simeq S^{n-2}$. ▯

In the remainder of this section we will study the structure of intervals in the lattice $\mathsf{Pos}_n^\#$. This has no immediate relevance for the topology of graph properties; thus, the rest of this section can be skipped without loss of continuity. Let us mention, however, that a different proof of Theorem 2.2 is given as a special case of the proof of Theorem 2.6 below.

We could not find any reference in the literature to the full lattice $\mathsf{Pos}_n^\#$ of all posets on $[n]$, although such a fundamental object must surely have been considered before. However, its lower intervals $[\hat{0}, P]$ for $P \neq \hat{1}$ have been studied by Edelman and Klingsberg [7] and others before them; see the references in [7]. A lattice of the form $[\hat{0}, P]$ is the lattice of all subposets of a given poset P. It is known that such lattices are *meet-distributive*, meaning that if x is the meet of all elements covered by y, then the interval $[x, y]$ is Boolean. From this can be deduced that the order complex of each open interval (Q, P), $P \neq \hat{1}$, is either contractible or a sphere, and the Möbius function takes values $\mu(Q, P) \in \{0, \pm 1\}$ for $P \neq \hat{1}$.

We will extend this topological description of open intervals to *all* intervals of $\mathsf{Pos}_n^\#$ and make it more explicit in the known cases.

We call a poset $(P, \leq_P) \in \mathsf{Pos}_n^\#$ a *Coxeter poset* if it is the antichain (i.e., the bottom element $\hat{0}$) or if there is a point $(x_1, \ldots, x_n) \in X_n = \{\underline{\mathbf{x}} \in \mathbb{R}^n \mid x_1 + \cdots + x_n = 0 \text{ and } \|\underline{\mathbf{x}}\| = 1\}$ such that $i <_P j$ if and only if $x_i < x_j$. It follows that i and j are incomparable in P if and only if $x_i = x_j$. We say that the point $(x_1, \ldots, x_n)$ *realizes* P.

LEMMA 2.3. *For a poset* $(P, \leq_P) \in \mathsf{Pos}_n \cup \{\hat{0}\}$ *the following are equivalent:*

(i) *P is a Coxeter poset.*

(ii) *No element in P is incomparable to both elements of some* 2*-element chain.*

(iii) *P is an ordinal sum of antichains.*

Proof. Assume i is incomparable to $j <_P k$. Then P is not a Coxeter poset, since otherwise $x_i = x_j < x_k = x_i$ for the coordinates of every point that realizes P.

We have shown that (i) $\Rightarrow$ (ii), and (ii) $\Rightarrow$ (iii) follows from an easy combinatorial argument.

Suppose that $P = A_1 \oplus A_2 \oplus \cdots \oplus A_k$, where the A_i's denote antichains on the respective blocks A_i of a partition of the set $[n]$. If $k = 1$ assertion (i) follows trivially. Assume $k \geq 2$. Define a point $\underline{\mathbf{y}} = \{y_1, \ldots, y_n\} \in \mathbb{R}^n$ by setting $y_i = j$ if $i \in A_j$, and let $N = \sum_1^n y_i$. Then let $\underline{\mathbf{z}}$ be the point obtained by subtracting N/n from each coordinate of $\underline{\mathbf{y}}$, and finally let $\underline{\mathbf{x}}$ be the vector $\underline{\mathbf{z}}$ normalized to length one. One easily checks that $\underline{\mathbf{x}} \in X_n$ and that $\underline{\mathbf{x}}$ realizes P. Hence, P is a Coxeter poset. □

For example, the Coxeter posets in Pos_3 are the twelve elements on the two top rank levels; see Figure 2.1.

In the following we denote by Cox_n the *Coxeter complex* of the symmetric group Σ_n. The Coxeter complex is constructed as follows. For each $p \in X_n$ let $\overline{p}$ be the intersections of all closed hemispheres of the form $\{\underline{\mathbf{x}} \in X_n \mid x_i \leq x_j\}$ that contain p. The sets of the form $\overline{p}$ are the simplices of a triangulation of the $(n-2)$-sphere X_n called the *Coxeter complex* Cox_n; see, e.g., [5, Chapter 2.3]. We describe an inclusion map from the face poset of Cox_n to Pos_n.

LEMMA 2.4. *The subposet of Pos_n consisting of all Coxeter posets is isomorphic to the proper part of the face lattice of the Coxeter complex Cox_n.*

Proof. This follows immediately from Lemma 2.3 and its proof. □

A Coxeter poset can easily be identified with a chain in the Boolean lattice B_n. Namely, if the poset is the ordinal sum $A_1 \oplus A_2 \oplus \cdots \oplus A_k$, where the A_i's denote antichains, the corresponding chain in B_n is $A_1, A_1 \cup A_2, A_1 \cup A_2 \cup A_3, \ldots$. Thus, the face lattice of Cox_n is isomorphic to the order complex of the proper part $\overline{B_n}$ of the Boolean lattice, or, to the barycentric subdivision of the boundary of an $(n-1)$-simplex. This is a well-known combinatorial description of Cox_n.

We will call a poset $(P, \leq_P)$ a *chainbreaker* for a poset $(Q, \leq_Q)$ if $P < Q$ and for at least one chain $i <_Q j <_Q k$ in Q the element i is incomparable to k in P. The second part of the following lemma expresses the meet-distributivity of lower intervals; see also [7].

LEMMA 2.5. *Let $[P, Q]$ be an interval in $\mathsf{Pos}_n^{\#}$, with $Q \neq \hat{1}$. If P is a chainbreaker for Q, then the open interval (P, Q) is contractible. Otherwise $[P, Q]$ is isomorphic to the Boolean lattice on a* $\operatorname{rank}(Q) - \operatorname{rank}(P)$ *element set.*

Proof. Let $i <_Q j <_Q k$ be a chain in Q such that i is incomparable to k in P. The order relation $i < k$ will be present in any poset covered by Q. In particular, in the meet of all coatoms of $[P, Q]$ we will have the order relation $i < k$. Thus P is not the meet of these coatoms. But this shows that the crosscut complex $\Gamma([P, Q]^*, A)$ of the dual lattice (i.e., all order relations reversed) of $[P, Q]$ is the full simplex over the set A of coatoms of $[P, Q]$. In particular, it is contractible.

Now assume that P is not a chainbreaker for Q. Let $i <_Q k$ be an order relation in Q such that i and k are incomparable in P. Let $(R, \leq_R)$ be the poset which inherits all order relations from Q except $i <_Q k$. We have to show that R is indeed a poset. Only the transitivity of the order relation is not completely trivial. But it follows from the observation that there are no chains $i <_Q j <_Q k$ in Q, since otherwise P would be a chainbreaker for Q. Thus we can remove an arbitrary order relation from $\operatorname{Rel}(Q) \setminus \operatorname{Rel}(P)$ and obtain a poset R covered by Q. Since P is not a chainbreaker for R the assertion follows by induction. □

We are now ready to formulate a theorem describing all intervals of $\mathsf{Pos}_n^{\#}$ up to homotopy type. The *height* of a poset P is by definition the maximal number k such that P contains some chain $y_0 < y_1 < \cdots < y_k$.

THEOREM 2.6. *Let (P, Q) be an open interval in $\mathsf{Pos}_n^{\#}$.*

(i) *If either $Q \neq \hat{1}$ and P is a chainbreaker for Q or $Q = \hat{1}$ and P is not a Coxeter poset, then the interval (P, Q) is contractible.*

(ii) *If $Q \neq \hat{1}$ and P is not a chainbreaker for Q, then (P, Q) is homeomorphic to a sphere of dimension* $\mathrm{rank}(Q) - \mathrm{rank}(P) - 2$.

(iii) *If $Q = \hat{1}$ and P is a Coxeter poset, then (P, Q) is homotopy equivalent to a sphere of dimension $n - 2 - \mathrm{height}(P)$.*

It follows as a special case of part (iii) that $\mathsf{Pos}_n = (\hat{0}, \hat{1})$ is homotopy equivalent to the $(n-2)$-dimensional sphere, so we obtain below a new proof for Theorem 2.2.

Proof. Lemma 2.5 settles the case $Q \neq \hat{1}$. It therefore remains to show the following for the case $Q = \hat{1}$:

(a) If P is not a Coxeter poset, then the interval $(P, \hat{1})$ is contractible.

(b) If P is a Coxeter poset, then $(P, \hat{1})$ is homotopy equivalent to a sphere of dimension $n - 2 - \mathrm{height}(P)$.

All assertions are trivial for $n = 1$, so we assume $n \geq 2$.

By Lemma 2.3 we know that if $(P, \leq_P)$ is not a Coxeter poset, then there is an element j that is incomparable to a 2-element chain $i <_P k$. Now consider the poset $(R, \leq_R)$ which is the meet of all linear extensions of P such that $i <_P j <_P k$ (such extensions exist!). Clearly, $i <_R j <_R k$ in R and hence $(R, \leq_R) \neq (P, \leq_P)$. We claim that there is no poset $(S, \leq_S)$ in $(P, \hat{1})$ that complements R (i.e., such that the meet of S and R is P and their join is $\hat{1}$). Let us distinguish three cases:

- "$j <_S k$": Then j is smaller than k in the meet of S and R. In particular, the meet is not P.
- "$i <_S j$": Then j is larger than i in the meet of S and R. In particular, the meet is not P.
- "j is incomparable to $i <_S k$ in S": Then there is a linear extension $(T, \leq_T)$ of S such that $i \leq_T j \leq_T k$. But then $T \geq R$ and the join of S and R is not $\hat{1}$.

By Proposition 7.3 it follows that $(P, \hat{1})$ is contractible.

For part (b) we consider the inclusion map $f : \overline{L}(\mathsf{Cox}_n) \hookrightarrow \mathsf{Pos}_n$ from the face poset of Cox_n to Pos_n described in Lemma 2.4. We will use Quillen's fiber lemma, Proposition 7.1. Let $(P, \leq_P)$ be a poset in Pos_n and set $Q := f^{-1}((\mathsf{Pos}_n)_{\geq P})$.

Claim. Q is contractible.

Let $R := \overline{L}(\mathsf{Cox}_n) \setminus Q$, and let

$$X_P = \Big\{ (x_1, \ldots, x_n) \in X_n \Big| \; i <_P j \Rightarrow x_i < x_j \Big\}.$$

We have that X_P is an intersection of open hemispheres. From the fact that P is a poset it follows that X_P is nonempty. Thus X_P is contractible. On the other hand, $X \setminus X_P$ is triangulated by $\Delta(R)$, so via a retraction argument (see, e.g., [5, Lemma 4.7.27]) $\Delta(Q)$ and X_P are homotopy equivalent.

Thus the Quillen fiber lemma implies that, for all P in the image of f, the interval $(P, \hat{1})$ in $\mathsf{Pos}_n^{\#}$ is homotopy equivalent to the interval $(P, \hat{1})$ in $L(\mathsf{Cox}_n)$. An upper interval $(\sigma, \hat{1})$ in the face lattice of a simplicial complex is homeomorphic to the link of σ in the complex. Now, Cox_n is a PL triangulation of the $(n-2)$-sphere, so the link of a simplex of dimension c is homeomorphic to a sphere of dimension $n - 3 - c$.

It is easily seen that the dimension of P as a simplex of Cox_n is height$(P) - 1$. This completes the proof. ∎

Corollary 2.7. *The Möbius function of* $\mathsf{Pos}_n^{\#}$ *is for* $P < Q$ *given by*

$$\mu(P,Q) = \begin{cases} 0 & \text{if } Q \neq \hat{1} \text{ and } P \text{ is a chainbreaker for } Q, \\ (-1)^{\mathrm{rank}(Q)-\mathrm{rank}(P)} & \text{if } Q \neq \hat{1} \text{ and } P \text{ is not a chainbreaker for } Q, \\ (-1)^{n-\mathrm{height}(P)} & \text{if } Q = \hat{1} \text{ and } P \text{ is a Coxeter poset,} \\ 0 & \text{if } Q = \hat{1} \text{ and } P \text{ is not a Coxeter poset.} \end{cases}$$

3. Not strongly connected digraphs. Let G be a digraph on the node set $[n]$. Say that two nodes u and v are G-equivalent if there is a directed path from u to v and a directed path from v to u. This is clearly an equivalence relation, and hence determines a partition of the set of nodes into equivalence classes B_i. We will denote this partition by $\pi(G) = |B_1|\cdots|B_k|$. The induced subgraphs G_{B_i} on the node sets B_i are called the *strongly connected components* of G.

We have defined an order-preserving map π from the set of all digraphs on $[n]$ (ordered by inclusion of edge sets) to the partition lattice Π_n. Note that $\pi(G) = |1|\cdots|n|$ (the bottom element of Π_n) if and only if G is acyclic. Also, $\pi(G) = |1\cdots n|$ (the top element of Π_n) if and only if G is strongly connected.

Let $\mathsf{Pos}_n \oplus \overline{\Pi_n}$ be the ordinal sum of Pos_n and the proper part $\overline{\Pi_n}$ of the partition lattice Π_n, and define a map $\varphi : \Delta_n^{NSC} \setminus \{\emptyset\} \to \mathsf{Pos}_n \oplus \overline{\Pi_n}$ by

$$\varphi : G \mapsto \begin{cases} \widetilde{G} \in \mathsf{Pos}_n & \text{if } G \text{ is acyclic,} \\ \pi(G) \in \overline{\Pi_n} & \text{otherwise.} \end{cases}$$

Thus, φ sends a not strongly connected digraph G to its transitive closure $\widetilde{G}$ in case $\pi(G)$ is the bottom element of Π_n and to the partition $\pi(G)$ otherwise. This mapping is clearly order-preserving, and from now on we think of it as a poset map defined on the face poset of Δ_n^{NSC}.

Lemma 3.1. *The poset mapping* $\varphi : \overline{L}(\Delta_n^{NSC}) \to \mathsf{Pos}_n \oplus \overline{\Pi_n}$ *induces homotopy equivalence of order complexes.*

Proof. We will use Quillen's fiber lemma, Proposition 7.1. To simplify notation, let $Q := \mathsf{Pos}_n \oplus \overline{\Pi_n}$ for the duration of this proof. Let $q \in Q$. We have to show that the fiber $\varphi^{-1}(Q_{\le q})$ is contractible. There are two cases to consider.

Case 1. $q \in \mathsf{Pos}_n$. Here the fiber $\varphi^{-1}(Q_{\le q})$ has a unique greatest element, namely, the comparability graph of the poset q. So its order complex is a cone and hence contractible.

Case 2. $q \in \overline{\Pi_n}$. We will use the $k = 2$ case of Lemma 7.7. Now q is a nontrivial partition of the node set $[n]$. Assume that B is a nonsingleton block in this partition and choose two elements in B. Without loss of generality we may assume that the two elements are 1 and 2.

The fiber $\Delta := \varphi^{-1}(Q_{\le q})$ is the subcomplex of Δ_n^{NSC} consisting of those digraphs G such that $\pi(G) \le q$. Let $\Delta_1 \subseteq \Delta$ be the subcomplex consisting of those graphs in Δ that contain no simple directed path from 1 to 2 except possibly for the edge $(1,2)$. The map $G \mapsto G\pm(2,1)$ (defined in connection with Lemma 7.7) maps Δ_1 into itself. If $(2,1) \in G$ this is clear. If $(2,1) \notin G$, then $\pi(G\pm(2,1))$ might differ from $\pi(G)$ by the merging of the blocks B_1 and B_2 containing 1 and 2. But then we still have that $\pi(G\pm(2,1)) \le q$, since B_1 and B_2 are both subsets of B. Similarly, $G \mapsto G\pm(1,2)$ maps $\Delta \setminus \Delta_1$ into itself, because $\pi(G\pm(1,2)) = \pi(G)$. Hence, by Lemma 7.7, Δ is contractible. ∎

The following result will, together with Lemma 3.1, imply the truth of Theorem 1.2.

LEMMA 3.2. *$\Delta(\mathsf{Pos}_n \oplus \overline{\Pi_n})$ has the homotopy type of a wedge of $(n-1)!$ spheres of dimension $2n-4$.*

Proof. From the definition of ordinal sum follows that $\Delta(\mathsf{Pos}_n \oplus \overline{\Pi_n}) = \Delta(\mathsf{Pos}_n) * \Delta(\overline{\Pi_n})$, where "$*$" denotes join of simplicial complexes. It is a known fact that if Δ_1 is homotopy equivalent to a wedge of k_1 spheres of dimension d_1 and Δ_2 is homotopy equivalent to a wedge of k_2 spheres of dimension d_2, then their join $\Delta_1 * \Delta_2$ is homotopy equivalent to a wedge of $k_1 \cdot k_2$ spheres of dimension $d_1 + d_2 + 1$ (see, for example, [6, Lemma 2.5]). It is also well known that the proper part $\overline{\Pi_n}$ of the partition lattice has the homotopy type of a wedge of $(n-1)!$ spheres of dimension $n-3$; see, e.g., [6]. Also by Theorem 2.2 we know that Pos_n is homotopy equivalent to an $(n-2)$-sphere. Hence $\Delta(\mathsf{Pos}_n \oplus \overline{\Pi_n})$ is homotopy equivalent to a wedge of $(n-1)!$ spheres of dimension $(n-2)+(n-3)+1 = 2n-4$. ∎

4. Directed matchings. An undirected graph is called a matching if the degree (number of incident edges) at every node is at most 1. Similarly, we define a *directed matching* to be a digraph for which both the in-degree and out-degree at every vertex is at most one. So, the components of a directed matching are either directed paths or directed cycles.

Let Δ_n^{DM} be the simplicial complex of directed matchings on the node set $[n]$. This complex, whose set of vertices is Ω, can be described differently as follows.

Let A be a finite set of squares on a large enough chessboard. The *chessboard complex* $\mathcal{C}_A$ is the simplicial complex whose vertex set is A and whose simplices are the subsets of A corresponding to nontaking rook positions (i.e., no two squares in the same row or in the same column). Such complexes have been studied in several papers; see [1, 14] and the further references given there.

The following is immediately clear by identifying the edges (i,j) of Ω with the corresponding squares (i,j) of the $n \times n$ chessboard.

LEMMA 4.1. *Let A be the $n \times n$ chessboard minus one diagonal. Then $\mathcal{C}_A \cong \Delta_n^{DM}$.*

Using this lemma and a result of Ziegler [14] we can deduce this connectivity lower bound.

THEOREM 4.2. *The complex Δ_n^{DM} is $(\lfloor \frac{2n+1}{3} \rfloor - 2)$-connected.*

Proof. This is a consequence of [14, Theorem 3.3]. That theorem says that if a chessboard A contains a certain "admissible k-shape" $\Sigma(m,n,k)$, then the $(k-1)$-skeleton of its complex $\mathcal{C}_A$ is vertex-decomposable of dimension $k-1$, and hence, in particular, is $(k-2)$-connected. Now the $n \times n$ chessboard minus one diagonal contains an isomorphic copy of the admissible k-shape $\Sigma(n,n,k)$ for $k = \lfloor \frac{2n+1}{3} \rfloor$. ∎

It is easy to see that the $(n-1)$-dimensional complex Δ_n^{DM} collapses to its $(n-2)$-skeleton. Hence we may deduce the following vanishing result:

$$\widetilde{H}_i(\Delta_n^{DM}) \neq 0 \quad \Longrightarrow \quad \left(\left\lfloor \frac{2n+1}{3} \right\rfloor - 1 \right) \leq i \leq n-2.$$

These bounds are sharp for $n \leq 7$, as Table 4.1 shows.

The only torsion appearing in the table is modulo 3. Thus the directed matching complexes Δ_n^{DM} seem to share the mysterious "torsion mod 3" phenomenon empirically observed for undirected matching complexes; see the discussion in [1, Section 9.1].

TABLE 4.1
Homology groups $\widetilde{H}_i(\Delta_n^{DM})$.

$n \backslash i$	0	1	2	3	4	5	6
2	0	0	0	0	0	0	0
3	0	$\mathbb{Z}^2$	0	0	0	0	0
4	0	0	$\mathbb{Z}^4$	0	0	0	0
5	0	0	$\mathbb{Z}$	$\mathbb{Z}^{13}$	0	0	0
6	0	0	0	$\mathbb{Z}^{24} \oplus \mathbb{Z}_3^5$	$\mathbb{Z}^{32}$	0	0
7	0	0	0	0	$\mathbb{Z}^{415} \oplus \mathbb{Z}_3^{15}$	$\mathbb{Z}^{95}$	0

5. Nonspanning digraphs and arborescences. In this section we will first consider a digraph property that is defined with respect to a root node. Since only outgoing edges from the root play a role, this introduces a small asymmetry into the ground sets of nodes and of edges. Thus we enlarge our standard ground sets $[n]$ to $[n]_0 = [n] \cup \{0\}$ and Ω (defined in section 1) to $\Omega_0 = \Omega \cup \{(0, j) \mid 1 \le j \le n\}$. This means simply that we introduce 0 as the root node and add edges from 0 to all other nodes j.

A rooted digraph is called *spanning* (or, *initially connected*) if it contains a directed path from the root 0 to every other node i. The nonspanning digraphs form a simplicial complex on the vertex set Ω_0 that we denote Δ_n^{NS}.

THEOREM 5.1. *The complex* Δ_n^{NS} *is contractible.*

Proof. Let $f : \overline{L}(\Delta_n^{NS}) \to B_n$ be the map that assigns to each digraph in $\overline{L}(\Delta_n^{NS})$ the subset of $[n]$ consisting of all nodes that can be reached from 0 on a directed path. Clearly, this map is order preserving and its image consists of all subsets of $[n]$ of cardinality $\le n-1$. In particular, the order complex of $f(\overline{L}(\Delta_n^{NS}))$ is contractible, since it is a cone with the empty set as apex. Thus we are done once we show that f induces a homotopy equivalence. For this we apply the Quillen fiber lemma, Proposition 7.1.

For $j = 0, 1, \ldots, n-1$, let $\Delta_j := \{G \in \Delta_n^{NS} | f(G) \subseteq [j]\}$. Thus, Δ_j consists of all digraphs such that only a subset of $[j] = \{1, \ldots, j\}$ can be reached along a directed path from 0. We have that $\Delta_0 \subseteq \Delta_1 \subseteq \cdots \subseteq \Delta_{n-1}$ and plan to use Lemma 7.7.

First observe that Δ_0 is the full simplex on the vertex set Ω, and hence is a cone. Then, notice that $G \mapsto G \pm (j+1, j)$ maps $\Delta_j \setminus \Delta_{j-1}$ into itself for all $1 \le j \le n-1$. Hence, by Lemma 7.7, every complex Δ_j is contractible.

The argument is now finished, since by symmetry the Quillen fiber $f^{-1}((B_n)_{\le A})$ is isomorphic to $\Delta_{|A|}$ for every subset A of $[n]$ such that $|A| \le n-1$. ∎

Another digraph property we will consider here is that of being an *arborescence* (or, *directed forest*). This means that each connected component is a directed tree, i.e., a tree having a node r such that every other node of the tree can be reached from r along a directed path. In other words, every component of an arborescence is a tree in which all edges are directed "away" from some particular node. Being an arborescence is clearly a hereditary property. We denote by Δ_n^{AR} the complex of arborescences. It is a simplicial complex on the vertex set Ω_0.

Relying on some results on greedoids by Björner, Korte, and Lovász [4] we can deduce the following.

THEOREM 5.2. *The complex* Δ_n^{AR} *is shellable and contractible.*

Proof. We will here freely use greedoid terminology. See [4] for all definitions.

Consider the branching greedoid G of the complete digraph Ω_0 rooted at 0. The feasible sets of G are the directed trees rooted at 0, and the bases are the maximal

Table 6.1
Homology groups $\widetilde{H}_j(\Delta_n^{NSC,2})$.

$n\backslash j$	0	1	2	3	4	5	6	7	8	9	10
3	0	0	0	0	$\mathbb{Z}$	0	0	0	0	0	0
4	0	0	0	0	0	0	0	$\mathbb{Z}^4$	0	0	0
5	0	0	0	0	0	0	0	0	0	0	$\mathbb{Z}^{18}$

arborescences (by necessity rooted at 0). Hence, Δ_n^{AR} is the primal complex of G, which by [4, Theorem 5.8] is shellable and by [4, Theorem 5.9] is contractible. □

For $n \geq 4$ Theorem 5.1 can also be proved using methods from greedoid theory.

Proof (greedoid proof of Theorem 5.1). Once more, let G be the branching greedoid of the complete digraph Ω_0 rooted at 0. A digraph D is spanning if and only if D contains a basis of G, which is same as saying that the complement $\Omega_0 \setminus D$ belongs to the dual complex $G^{\perp}$. Hence, Δ_n^{NS} is Alexander dual to $G^{\perp}$ in the sense of Proposition 7.5, which therefore gives

$$\widetilde{H}_i(\Delta_n^{NS}) \cong \widetilde{H}^{n^2-i-3}(G^{\perp}).$$

By [4, Corollary 5.12] $G^{\perp}$ is homeomorphic to a ball. Hence, $\widetilde{H}_i(\Delta_n^{NS}) = 0$ for all i.

If $n \geq 4$, then Δ_n^{NS} has a complete 2-skeleton, so Δ_n^{NS} is simply-connected in this case and therefore contractible. □

Remark 5.3. The complex of arborescences has been studied also by Kozlov [8]. He proves shellability both for Ω_0 (as in Theorem 5.2) and for the smaller ground set Ω.

Remark 5.4. The argument by Alexander duality in the last proof can be extended to all greedoids. Let G be a rank r greedoid on a ground set of cardinality m. By [4, Theorem 5.1] the dual complex $G^{\perp}$ is shellable. Since it is also pure and of dimension $m-r-1$ it follows that $\widetilde{H}^j(G^{\perp}) = 0$ for all $j \neq m-r-1$ and $\widetilde{H}^{m-r-1}(G^{\perp})$ is free. The complex NS of nonspanning sets of G, which is Alexander dual to $G^{\perp}$, therefore satisfies $\widetilde{H}_i(NS) = 0$ for all $i \neq m-(m-r-1)-3 = r-2$, and $\widetilde{H}_{r-2}(NS)$ is free. If $r \geq 4$, then NS has a complete 2-skeleton and is therefore simply-connected. From these facts we may conclude (see also [3, (9.18)]) that NS has the homotopy type of a wedge of $(r-2)$-spheres.

A particular case of this is given by the complex of disconnected undirected graphs on the vertex set $[n]$, denoted by Δ_n^1 in [1]. This is the complex of nonspanning sets in the circuit matroid of the complete graph K_n, a greedoid of rank $n-1$. It is well known that Δ_n^1 has nonvanishing homology only in dimension $n-3$; see, e.g., [1, Proposition 2.1].

6. Final remarks.

6.1. Strongly *i*-connected digraphs. Following the program pursued in [1] one can extend the considerations of section 3 to the complex $\Delta_n^{NSC,i}$ of not strongly i-connected digraphs on the node set $[n]$. We will say that a digraph $(V(G), E(G))$ is *strongly i-connected*, for a number $0 < i < |V(G)|$, if for any j vertices $v_1, \ldots, v_j$, $j < i$, the digraph that is obtained from G by removing $v_1, \ldots, v_j$ and all incident edges is strongly connected. In particular, $\Delta_n^{NSC} = \Delta_n^{NSC,1}$. Computer calculations for $i = 2$ yield the homology groups shown in Table 6.1 (no nonvanishing homology occurs for j not in the table).

Based on this table one is led to suspect the following.

CONJECTURE 6.1. $\Delta_n^{NSC,2}$ *is homotopy equivalent to a wedge of* $(n-2)!(n-2)$ *spheres of dimension* $3n-5$.

Now we consider situations when i is close to n. A digraph is not strongly $(n-1)$-connected if there is a pair of vertices such that the subgraph induced on these vertices is not strongly connected. But this is equivalent to the condition that the graph is not the complete digraph on $[n]$ vertices. Thus, $\Delta_n^{NSC,n-1}$ is the full boundary of an (n^2-n-1)-simplex, and hence $\Delta_n^{NSC,n-1} \cong S^{n^2-n-2}$.

In order to study $\Delta_n^{NSC,n-2}$ we consider the dual complex in the sense of Proposition 7.5. It is easily derived from the definitions that a digraph lies in $(\Delta_n^{NSC,n-2})^*$ if and only if the out-degree and in-degree at each node are at most 1. Hence, by Proposition 7.5,

$$\widetilde{H}_i(\Delta_n^{NSC,n-2}) \cong \widetilde{H}^{n^2-n-i-3}(\Delta_n^{DM}),$$

and the results of section 4 apply.

6.2. Action of the symmetric group. For each monotone property P of digraphs on n nodes, the complex Δ_n^P is invariant under the action of the symmetric group Σ_n by permutations of the node set n. This action induces a Σ_n-representation on each homology group $\widetilde{H}_i(\Delta_n^P;\mathbb{C})$. For Δ_n^{ACY} and Δ_n^{NSC} we can describe these representations.

Essentially, what has to be done is to keep track that all the homotopy equivalences established in sections 2 and 3 are compatible with the group action. This can be done using an equivariant version of the Quillen fiber lemma due to Thévenaz and Webb [12] and the equivariant versions of crosscut theorem and homotopy complementation that can be derived from it. Using this strategy the following results can be obtained:

- The complex Δ_n^{ACY} is Σ_n-homotopy equivalent to the Coxeter complex Cox_n.
- The complex Δ_n^{NSC} is Σ_n-homotopy equivalent to the join of Cox_n and $\Delta(\overline{\Pi_n})$.

It is easily seen and well known that on the single nonvanishing homology group $\widetilde{H}_{n-2}(\mathsf{Cox}_n;\mathbb{C})$ of Cox_n the group Σ_n acts by the sign-character $\mathbf{sign}_n$. By results of Stanley [11] the character of Σ_n on the only nonvanishing homology group $\widetilde{H}_{n-3}(\Delta(\overline{\Pi_n});\mathbb{C})$ of $\Delta(\overline{\Pi_n})$ is given by $\mathbf{sign}_n \cdot \mathbf{lie}_n$. Here $\mathbf{lie}_n := \mathbf{e}^{\frac{2\pi\mathbf{i}}{\mathbf{n}}} \uparrow_{\mathbf{C_n}}^{\mathbf{\Sigma_n}}$, where C_n denotes the cyclic group of order n generated by an n-cycle and $\mathbf{e}^{\frac{2\pi\mathbf{i}}{\mathbf{n}}}$ the character of C_n which assumes the value $e^{\frac{2\pi i}{n}}$ on a generator of the group. Using these two facts and the Σ_n-homotopy equivalences stated above we immediately obtain the following:

- The character of Σ_n on $\widetilde{H}_{n-2}(\Delta_n^{ACY};\mathbb{C})$ is the sign-character.
- The character of Σ_n on $\widetilde{H}_{2n-4}(\Delta_n^{NSC};\mathbb{C})$ is the character $\mathbf{lie}_n$.

We remark that $\mathbf{lie}_n$ is also the character of Σ_n on the multigraded part of the free Lie algebra on n generators (see, e.g., [10]).

7. Appendix: Notation and tools. In this section we will summarize the main tools used. We refer the reader to the survey paper [3] for more details and references.

Let P be a finite poset. If P has a unique minimum element $\hat{0}$ and a unique maximum element $\hat{1}$, we denote by $\overline{P}$ the *proper part* of P; that is, the poset obtained by removing from P the elements $\hat{0}$ and $\hat{1}$. By $\Delta(P)$ we denote the simplicial complex of all chains in P, called the *order complex* of P. Via the functor $\Delta(\cdot)$ one can speak of the homology and homotopy type of a poset P.

By convention we include the empty set $\emptyset$ in every simplicial complex. For any simplicial complex Δ, the *face lattice* $L(\Delta)$ is the poset of faces of Δ, ordered by inclusion and enlarged by an additional greatest element $\hat{1}$. The proper part $\overline{L}(\Delta) = L(\Delta) \setminus \{\emptyset, \hat{1}\}$ is called the *face poset*. The order complex $\Delta(\overline{L}(\Delta))$ is homeomorphic to Δ — indeed, $\Delta(\overline{L}(\Delta))$ is the barycentric subdivision of Δ.

The *ordinal sum* $P \oplus Q$ of two posets P and Q is the poset on their set union for which the order relation on pairs of elements of P (resp., Q) is inherited, and each element of P is defined to be less than each element of Q. This operation on posets is associative, so repeated ordinal sums $P_1 \oplus P_2 \oplus \cdots \oplus P_k$ are well defined.

For a poset P and $p \in P$ we denote by $P_{\leq p}$ the subposet $\{p' \mid p' \in P;\ p' \leq p\}$, and similarly for $P_{\geq p}$. For $p \leq p'$ in P we denote by $[p, p']$ the *closed interval* $P_{\geq p} \cap P_{\leq p'}$, and by (p, p') the *open interval* $[p, p'] \setminus \{p, p'\}$. By a map $f : P \to Q$ of posets we always mean a poset homomorphism (i.e., $x \leq y$ implies $f(x) \leq f(y)$).

Proposition 7.1 (Quillen fiber lemma (see [9], [3, Thm. 10.5])). *Let $f : P \to Q$ be a map of posets. Assume that $f^{-1}(Q_{\leq q})$ is contractible for all $q \in Q$. Then $\Delta(P)$ and $\Delta(Q)$ are homotopy equivalent.*

A map $f : P \to P$ from a poset to itself is called a *closure operator* if $f(x) \geq x$ and $f(f(x)) = f(x)$ for all $x \in P$. The Quillen fiber lemma immediately implies the following fact.

Corollary 7.2 (closure lemma). *Let $f : P \to P$ be a closure operator on the partially ordered set P. Then $\Delta(P)$ and $\Delta(f(P))$ are homotopy equivalent.*

A poset L is called a *lattice* if suprema, denoted by "$\vee$", and infima, denoted by "$\wedge$", exist. Note that if L is a finite lattice, then there is a least element $\hat{0}$ and a greatest element $\hat{1}$ in L. The elements covering $\hat{0}$ are called *atoms* and the elements covered by $\hat{1}$ are called *coatoms*. In a lattice an element x is called *complement* of the element y if $x \vee y = \hat{1}$ and $x \wedge y = \hat{0}$.

Proposition 7.3 (homotopy complementation (see [2], [3, Thm. 10.15])). *If some element of a lattice L has no complement, then $\Delta(\overline{L})$ is contractible.*

Let A be the set of atoms of a lattice L. Define the *crosscut complex* $\Gamma(L, A)$ to be the simplicial complex on the vertex set A whose simplices are the subsets $S \subseteq A$ such that $\vee S < \hat{1}$.

Proposition 7.4 (crosscut theorem (see [2], [3, Thm. 10.8])). *The complexes $\Gamma(L, A)$ and $\Delta(\overline{L})$ are homotopy equivalent.*

Our next tool is the combinatorial version of a standard duality theorem from algebraic topology. See, e.g., [1, Prop. 10.4] for a proof.

Proposition 7.5 (combinatorial Alexander duality). *Let Δ be a finite simplicial complex on vertex set V and define*

$$\Delta^* = \{B \subseteq V \mid V \setminus B \notin \Delta\}.$$

Then

$$\widetilde{H}_i(\Delta) \cong \widetilde{H}^{|V|-i-3}(\Delta^*).$$

Finally, we make use of the following simple collapsing argument. Assume there is a fixed ground set V and for $F \subseteq V$ and $a \in V$ define the operation

$$F \pm a = \begin{cases} F \cup \{a\} & \text{if } a \notin F, \\ F \setminus \{a\} & \text{if } a \in F. \end{cases}$$

Lemma 7.6. *Let $\Delta_1 \subseteq \Delta_2$ be simplicial complexes. Assume there exists some vertex a such that $F \mapsto F \pm a$ maps $\Delta_2 \setminus \Delta_1$ into itself. Then Δ_2 collapses to Δ_1.*

Proof. Take a pair $(F, F \pm a)$, $a \notin F \in \Delta_2 \setminus \Delta_1$, of maximal dimension among all such pairs. Suppose that F is not a free face. Then $F \cup \{b\} \in \Delta_2 \setminus \Delta_1$ for some

$b \notin F \cup \{a\}$. But since $a \notin F \cup \{b\}$, then also $F \cup \{b\} \cup \{a\} \in \Delta_2 \setminus \Delta_1$, contradicting the choice by maximal dimension. Hence, F is a free face, so the removal of $\{F, F \pm a\}$ is an elementary collapse step. Now continue by induction. ∎

The following generalization of the concept of a cone (the $k = 1$ case) is an immediate consequence.

LEMMA 7.7. *Let $\Delta_1 \subseteq \Delta_2 \cdots \subseteq \Delta_k = \Delta$ be simplicial complexes, and put $\Delta_0 = \emptyset$. Assume there exist vertices $a_1, a_2, \ldots, a_k$ such that $F \mapsto F \pm a_i$ maps $\Delta_i \setminus \Delta_{i-1}$ into itself, for $i = 1, \ldots, k$. Then Δ is collapsible.*

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] A. Björner, *Homotopy type of posets and lattice complementation*, J. Combin. Theory Ser. A, 30 (1981), pp. 90–100.
[3] A. Björner, *Topological Methods*, in Handbook of Combinatorics, R. Graham, M. Grötschel, and L. Lovász, eds., North-Holland, Amsterdam, 1995, pp. 1819–1872.
[4] A. Björner, B. Korte, and L. Lovász, *Homotopy properties of greedoids*, Adv. in Appl. Math., 6 (1985), pp. 447–494.
[5] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G. M. Ziegler, *Oriented Matroids*, Encyclopedia of Mathematics and Its Applications, Vol. 46, Cambridge University Press, Cambridge, UK, 1993.
[6] A. Björner and V. Welker, *The homology of "k–equal" manifolds and related partition lattices*, Adv. Math., 110 (1995), pp. 277–313.
[7] P. H. Edelman and P. Klingsberg, *The subposet lattice and the order polynomial*, European J. Combin., 3 (1982), pp. 341–346.
[8] D. Kozlov, *Complexes of Directed Trees*, preprint, 1998.
[9] D. Quillen, *Homotopy properties of the poset of non-trivial p-subgroups of a group*, Adv. Math., 28 (1978), pp. 101–128.
[10] C. Reutenauer, *Free Lie Algebras*, London Mathematical Society Monographs, New Series, Vol. 7, Oxford University Press, Oxford, UK, 1993.
[11] R. P. Stanley, *Some aspects of groups acting on finite posets*, J. Combin. Theory Ser. A, 32 (1982), pp. 132–161.
[12] J. Thévenaz and P. Webb, *Homotopy equivalence of posets with group action*, J. Combin. Theory Ser. A, 56 (1991), pp. 173–181.
[13] V. A. Vassiliev, *Complexes of connected graphs*, in The Gelfand Mathematical Seminar, 1990–1992, L. Corwin et al., eds., Birkhäuser, Boston, MA, 1993, pp. 223–235.
[14] G. M. Ziegler, *Shellability of chessboard complexes*, Israel J. Math., 87 (1994), pp. 97–110.

SIAM J. Discrete Math.
Vol. 12, No. 4, pp. 425-433

A 3/2-APPROXIMATION ALGORITHM FOR THE MIXED POSTMAN PROBLEM*

BALAJI RAGHAVACHARI† and JEYAKESAVAN VEERASAMY‡

Abstract. The mixed postman problem, a generalization of the Chinese postman problem, is that of finding the shortest tour that traverses each edge of a given mixed graph (a graph containing both undirected and directed edges) at least once. The problem is solvable in polynomial time either if the graph is undirected or if the graph is directed, but it is NP-hard in mixed graphs. An approximation algorithm with a performance ratio of 3/2 for the postman problem on mixed graphs is presented.

Key words. mixed postman problem, Chinese postman problem, NP-completeness, graphs, approximation algorithms

AMS subject classifications. 05C38, 05C45, 05C85, 68Q20, 68R10, 90B06, 90C27

PII. S0895480197331454

1. Introduction.

Problems of finding paths and tours on graphs are of fundamental importance and find many practical applications. The traveling salesman problem (TSP) is a well-known and widely studied problem. The objective is to find the shortest tour that visits all vertices of a given graph exactly once. The problem is known to be NP-hard. Postman problems are similar to the TSP at first glance but are quite different in terms of the complexity of the problems. Given a graph G, the Chinese postman problem (CPP) is to find a minimum-cost tour covering all edges of G at least once [10]. It is the optimization version of the Euler tour problem, which asks if there is a tour that traverses every edge of a graph exactly once. Edmonds and Johnson [4] showed that the problem is solvable in polynomial time. They also showed that the problem is solvable in polynomial time if G is a directed graph.

The *mixed postman problem* (MPP) is a generalization of the Chinese postman problem and is listed as problem ND25 by Garey and Johnson [9]. In the MPP, the input graph may contain both undirected edges and arcs (directed edges). The objective is to find a tour that traverses every edge at least once and that traverses directed edges only in the direction of the arc. Even though both undirected and directed versions of the CPP are polynomially solvable, Papadimitriou [15] showed that MPP is NP-hard. There are other related problems, such as the rural postman problem and the windy postman problem, which are also NP-hard [5, 6]. Many practical applications like mail delivery, snow removal, and trash pick-up can be modeled as instances of MPP, and hence it is important to design good approximation algorithms for this problem.

A key performance indicator of an approximation algorithm is its *approximation ratio*, which is the maximum ratio between the cost of the solution output by the

*Received by the editors December 11, 1997; accepted for publication (in revised form) April 6, 1999; published electronically October 19, 1999. The research of the first author was supported by NSF Research Initiation Award CCR-9409625.

http://www.siam.org/journals/sidma/12-4/33145.html

†Department of Computer Science, The University of Texas at Dallas, Richardson, TX 75083-0688 (rbk@utdallas.edu).

‡Department of Computer Science, The University of Texas at Dallas, Richardson, TX 75083-0688, and Samsung Telecommunications America, Inc., 1130 E. Arapaho Rd., Richardson, TX 75081 (veerasam@utdallas.edu).

algorithm and the cost of optimal solution over all input instances. An approximation algorithm with approximation ratio α is referred to as an α-approximation algorithm.

Previous work. Numerous articles have appeared in the literature over the past three decades about the MPP. Edmonds and Johnson [4] and Christofides [3] presented the first approximation algorithms. Frederickson [8] showed that the algorithm of [4] finds a tour whose length is at most two times the length of an optimal tour (i.e., approximation ratio of 2). He also presented a mixed strategy algorithm, which used the solutions output by two different heuristics, and then selected the shorter of the two tours. He proved that the approximation ratio of the mixed strategy algorithm is $\frac{5}{3}$. Comprehensive surveys are available on postman problems [2, 5]. Integer and linear programming formulations of postman problems have generated a lot of interest in recent years [11, 14, 18]. Ralphs [18] showed that a linear relaxation of MPP has optimal solutions that are half-integral. One could use this to derive a 2-approximation algorithm for the problem. Improvements in implementation are discussed in [14, 17]. It is interesting to note that Nobert and Picard [14] state that their implementation has been used for scheduling snow removal in Montreal. Several other articles have appeared on generalized postman problems, such as k-CPP [16] and the windy postman problem [5].

Our results. Even though numerous articles have appeared in the literature on MPP after Frederickson's paper in 1979, his result has been the best approximation algorithm for MPP in terms of proven worst-case ratio until now. In this paper, we present an improved approximation algorithm for MPP with an approximation ratio of $\frac{3}{2}$. We derive a new lower bound on the cost of an optimal solution using the properties of intermediate solutions to MPP. Our algorithm uses a subtle modification of Frederickson's algorithm, and the improved performance ratio is derived from the new lower bound.

2. Preliminaries. Problem statement. The input graph $G = (V, E, A)$ consists of a set of vertices V, a multiset of edges E, and a multiset of arcs (directed edges) A. A nonnegative cost function C is defined on edges and arcs. We extend the definition of C to multigraphs (multisets of edges and arcs) by taking the sum of the costs of its edges and arcs. The goal is to find the shortest tour that traverses all edges and arcs of G at least once. The tour may traverse edges in either direction, but must go through an arc only along its direction. We assume that the input graph is strongly connected, i.e., there exists a path from any vertex u to any other vertex v, since the problem is clearly infeasible for graphs that are not strongly connected. The output is a tour which may travel each edge or arc several times. Therefore, except for traversing each edge/arc once, the traversal could always use the shortest path between any two nodes.

Definitions. A *cut* is a partition of the vertex set into S and $V - S$. It is called *nontrivial* if neither side is empty. An edge *crosses* the cut if it connects a vertex in S to a vertex in $V - S$. For a vertex $v \in V$, let $outdegree(v)$ be the number of outgoing arcs from v. Similarly, $indegree(v)$ is the number of incoming arcs into v. Let $degree(v)$ be the total number of edges and arcs incident to v. We say v has even degree if $degree(v)$ is even. A graph has even degree if all its vertices have even degree. Let $surplus(v) = outdegree(v) - indegree(v)$. If $surplus(v)$ is negative, we may call it a *deficit*. The definition of surplus can be extended to sets of vertices S, by finding the difference between outgoing and incoming edges that cross the cut $(S, V - S)$. Given a graph and a subset of vertices, a T-join is defined as a subset of edges in which the degree of each vertex in the specified subset is odd and even for all other vertices. More formally, in a graph $G = (V, E)$ and $T \subseteq V$, a *T-join* is a subset of edges E' such that, in the subgraph $G' = (V, E')$, the degree of v is odd if and only if $v \in T$.

T-joins and matchings. In a connected graph $G = (V, E)$, there exists a T-join for any subset $T \subseteq V$ as long as the cardinality of T is even. A minimum-cost T-join can be computed using an algorithm for minimum-cost matchings on an auxiliary graph called the closure graph. The closure graph of G is a complete graph $H = (V, V \times V)$, where $c(u, v)$ is the cost of a shortest path between u and v. It can be shown that minimal T-joins in G correspond to matchings in H in which all nodes of T are matched. Refer to the book by Nemhauser and Wolsey [13] for more details on T-joins.

Properties of Eulerian graphs. A graph is called *Eulerian* if there is a tour that traverses each edge of the graph exactly once. It is known that an undirected, connected graph is Eulerian if and only if the degree of each vertex is even. For a directed graph to be Eulerian, the underlying graph must be connected, and for each vertex v, $outdegree(v) = indegree(v)$. In other words, for each vertex v, $surplus(v) = 0$. A mixed graph $G = (V, E, A)$ is Eulerian if the graph is strongly connected and satisfies the following properties [7, 14].

- *Even degree condition*: Every vertex v is incident to an even number of edges and arcs, i.e., $degree(v)$ is even. This condition implies that the number of edges and arcs crossing any cut $(S, V - S)$ is even.
- *Balanced set condition*: For every nontrivial cut $S \subset V$, the absolute value of the surplus of S must be less than or equal to the number of undirected edges crossing the cut $(S, V - S)$.

The above conditions can be checked in polynomial time using algorithms for the maximum flow problem [7]. In other words, we can decide in polynomial time whether a given mixed graph is Eulerian. The problem we are interested in is to find a set of additional edges and arcs of minimum total cost that can be added to G to make it Eulerian, and this problem is NP-hard. In the process of solving an instance of the mixed postman problem, arcs and edges may be duplicated. For convenience, we may also orient some undirected edges by giving them a direction. The output of our algorithm is a Eulerian graph H that contains the input graph G (in which some edges have been oriented) as a subgraph. So each edge of H can be classified either as an original edge or as a duplicated edge. Also, each arc of H is either an original arc, a duplicated arc, an oriented edge, or a duplicated and oriented edge.

3. Frederickson's Mixed algorithm.

Frederickson defined the following algorithms as part of his solution to MPP:

- EVENDEGREE: Augment a mixed graph G by duplicating edges and arcs such that the resulting graph has an even degree. A minimum-cost solution is obtained by disregarding the directions of the arcs (i.e., by taking the underlying undirected graph) and solving CPP by adding a minimum-weight T-join of all odd-degree nodes.
- INOUTDEGREE: Augment a mixed graph G by duplicating edges and arcs and orienting edges such that in the resulting graph, for each vertex v, $surplus(v) = 0$. We will refer to this as the INOUT problem. A minimum-cost solution G_{IO} is obtained by formulating a flow problem and solving it optimally. The flow problem is formulated such that the original undirected edges can be oriented free of cost. The *augmentation cost*, $C_{IO}(G)$, is defined as the cost of additional edges and arcs that are added to G to get G_{IO} by INOUTDEGREE.

In order to be self-contained, we briefly describe the min-cost flow formulation used by Frederickson [8] for the INOUT problem. Let $G = (V, E, A)$ be the input graph, where E is a set of undirected edges and A is a set of arcs. A flow network $N = (V, E')$ is created as follows. $E' = A \cup E_1 \cup E_2$, where E_1 and E_2 are defined as follows. For each edge $(v, w) \in E$, we add the arcs (v, w) and (w, v) to E_1 and also to E_2. The cost of an edge s is denoted by c_s. For a vertex v, let $b_v = -surplus(v)$ denote the imbalance created by the arcs of A.

$$\min z = \sum_{s \in A} c_s x_s + \sum_{s \in E_1} c_s x_s$$

subject to

$$x_s \leq 1 \text{ for all } s \in E_2, \tag{3.1}$$

$$Out(v) = \sum \{x_s | s \in E' \text{ is directed away from } v\} \text{ for all } v \in V, \tag{3.2}$$

$$In(v) = \sum \{x_s | s \in E' \text{ is directed towards } v\} \text{ for all } v \in V, \tag{3.3}$$

$$Out(v) - In(v) = b_v \text{ for all } v \in V, \tag{3.4}$$

$$x_s \geq 0 \text{ for all s} \in E'. \tag{3.5}$$

The arcs of E_2 are used to represent the fact that the edges of E can be oriented for free by the flow problem. Therefore, E_2 does not figure in the objective function, but the capacity of its edges is limited to 1 (constraint (3.1)). The arcs of E_1 represent the edges of E that are duplicated and oriented by the flow problem. Hence the variables corresponding to E_1 appear in the objective function. Each node v has an imbalance of b_v, which is defined as the number of arcs in A that are directed towards v minus the number of arcs in A that are directed away from v. Constraint (3.4) ensures that the solution to the flow problem corrects this imbalance at each vertex. It is not necessary to specify integrality conditions on the variables, since basic optimal solutions to flow problems are always integral. It can be shown that an integer optimal solution to the above linear program can be found using standard algorithms for the min-cost flow problem [1]. There is no loss of generality in assuming that, in an optimal solution to the above problem, both directed arcs corresponding to an undirected edge are not used in the solution. This is true because if a solution uses both arcs corresponding to an edge, then deleting both of these arcs retains feasibility of the solution without increasing the cost.

- EVENPARITY: When applied to the output of INOUTDEGREE on an even-degree graph, EVENPARITY restores even degree to all nodes without increasing the cost while retaining the property that *indegree* = *outdegree* at all nodes. Edmonds and Johnson [4] indicated that INOUTDEGREE can be applied to an even-degree graph in such a way that the resulting graph has even degree and is hence Eulerian. Frederickson [8] showed a simple linear-time algorithm to perform the task. The basis of this algorithm is that a suitably defined subgraph of undirected edges and duplicated edges/arcs forms a collection of Eulerian graphs.
- LARGECYCLES: LARGECYCLES is similar to EVENDEGREE except only edges are allowed to be duplicated, and arcs are not considered.

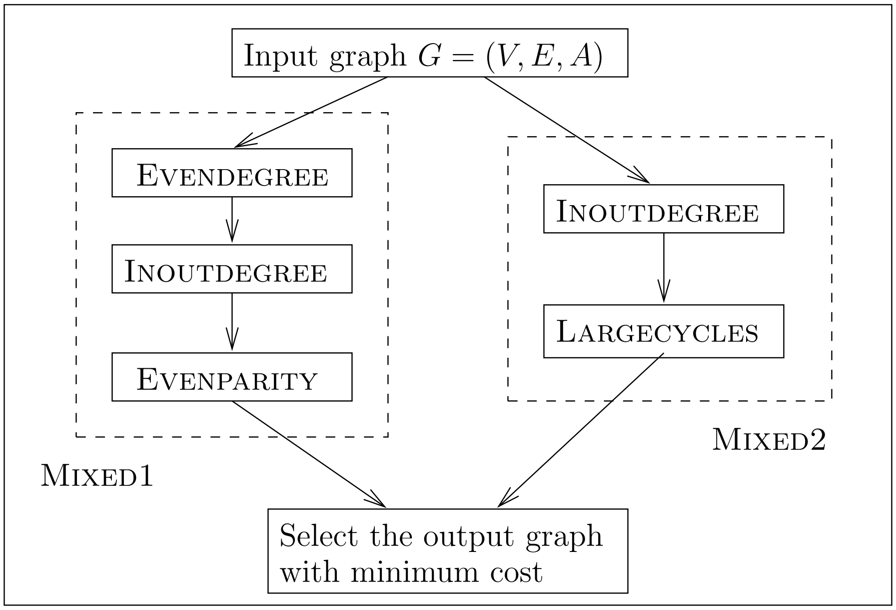


FIG. 1. MIXED *algorithm.*

Frederickson [8] presented an approximation algorithm for MPP called MIXED algorithm (see Figure 1). The algorithm comprises two heuristics called MIXED1 and MIXED2. Heuristic MIXED1 first runs EVENDEGREE to make the degree of all nodes even. Then it runs INOUTDEGREE to make *indegree* = *outdegree* for all vertices. Finally, EVENPARITY restores even degree to all the nodes without increasing the cost, and the graph becomes Eulerian. Heuristic MIXED2 first calls INOUTDEGREE and then makes the graph Eulerian by calling LARGECYCLES. Since LARGECYCLES does not duplicate arcs of the graph, no further steps are needed. The MIXED algorithm outputs the best solution of the two heuristics, and Frederickson showed that its performance ratio is at most $\frac{5}{3}$.

4. Improved lower bound for MPP. Consider a mixed graph $G = (V, E, A)$, with edges E and arcs A. Let C^* be the weight of an optimal postman tour of G. Suppose, given G as input, INOUTDEGREE outputs $G_{IO} = (V, U, M)$. $U \subseteq E$ are the edges of G that were not oriented by INOUTDEGREE. $M \supseteq A$ are arcs that satisfy *indegree* = *outdegree* at each vertex. For any edge $\{u, v\} \in E$, if M contains both arcs (u, v) and (v, u), remove both of these arcs and add the edge $\{u, v\}$ to U. This step does not affect the feasibility of the INOUT solution. Let C_M and C_U be the total weight of M and U, respectively. Note that $C_M + C_U = C_{IO}(G) + C(G)$. Consider the graph induced by M in the vertex set V. It contains one or more strongly connected components when M is not empty. Since all nodes in G_{IO} have surplus zero, each arc of M is in one of the strongly connected components. We will refer to such components as *directed components*. Let us define an equivalence relation Q on the nodes of G as follows: for any two nodes u and v, uQv if and only if u and v are in the same directed component in G_{IO}. It can be verified that Q is an equivalence relation. Suppose we shrink G by replacing the nodes of each equivalence class of Q by a single node. Let us denote the output multigraph as UG. Note that UG can also be derived from G_{IO} simply by shrinking each directed component to a single node. Since all arcs are part of directed components, there are no arcs in UG. We will refer to the nodes of UG as *supernodes* for clarity. Any node not adjacent to arcs in M is itself a directed component and hence a supernode of UG.

If all supernodes of UG are of even degree, we can find a T-join of all odd degree nodes of G by duplicating only the arcs and edges of M, and in this case it is possible to achieve an approximation ratio of $\frac{3}{2}$ using Frederickson's analysis. However, if there are supernodes of UG that have odd degree, i.e., if there are equivalence classes of Q with an odd number of odd degree nodes in them, then a T-join of odd-degree nodes is forced to use a subset of U. We need to improve the lower bound to account for these edges in order to achieve a $\frac{3}{2}$ approximation ratio for this case.

Consider a T-join J of the odd-degree nodes of G. The degree of each vertex in the union of J and G is even. Hence every cut in it is crossed by an even number of edges. Odd-degree supernodes in UG correspond to cuts in G that are crossed by an odd number of edges. Since the addition of J makes the number of edges crossing each of these cuts even, it can be verified that the set of edges of J that connect different supernodes forms a T-join of the odd-degree supernodes of UG. Also, if a T-join of the odd-degree supernodes of UG is added to G_{IO}, the number of odd-degree vertices in each of its directed components becomes even, and therefore the resulting graph can be made to have even-degree by duplicating edges of M alone. We show that Frederickson's lower bound on the optimal cost of MPP can be improved to include the cost of a minimum-weight T-join of the odd-degree supernodes of UG.

If we run EVENDEGREE on UG, it adds a minimum-cost T-join of odd-degree nodes (i.e., a subset of U) to make the degree of each supernode in UG even. Let the cost of this T-join be C_X. Frederickson [8] used a lower bound of $C_M + C_U$ on C^*. We show an improved lower bound on C^*, which allows us to improve the approximation ratio. We first show that when we find an optimal INOUT solution, if some undirected edges are not oriented (edges in the set U) by INOUTDEGREE, then adding additional copies of edges in U to G does not decrease $C_{IO}(G)$, the augmentation cost of INOUTDEGREE. The lemma is proved using LP duality. For more information on linear programs, the reader is referred to a book by Karloff [12].

LEMMA 4.1. *Let $G = (V, E, A)$ be a mixed graph. Let $G_{IO} = (V, U, M)$ be an optimal* INOUT *solution computed by* INOUTDEGREE *algorithm. Let G_{IO} be a minimal solution that has the property that it never uses both arcs that correspond to an undirected edge. Let $U = \{u_1, \ldots, u_k\}$ be the undirected edges of G that were not oriented by* INOUTDEGREE*, even though the algorithm could orient these edges without incurring additional cost. Adding additional copies of edges in U to G does not decrease the augmentation cost of an optimal* INOUT *solution.*

Proof. A linear program for the INOUT problem was presented in section 3. It is a min-cost flow problem whose objective function value is exactly the augmentation cost. Since the edges of U are not used in the optimal solution G_{IO}, they have a flow of 0 through them; i.e., the edges of U are unsaturated in G_{IO}. Adding additional copies of these edges is equivalent to increasing their capacities in the linear program. By linear program duality, increasing the capacity of unsaturated arcs does not affect the complementary slackness conditions, and hence an optimal solution to the linear program remains optimal even after the capacities of the arcs of U are increased. ∎

LEMMA 4.2. $C_M + C_U + C_X \leq C^*$.

Proof. Let G^* be an optimal solution, whose cost is C^*. Consider the supernodes of UG, which correspond to the directed components of M. G^* needs to have additional T-join edges (subset of U) between these components to satisfy the even-degree condition for each component. This T-join costs at least C_X. By Lemma 4.1, we know that additional edges of U do not decrease the cost of optimal INOUT solution. Therefore the augmentation cost of the INOUT problem is still $C_{IO}(G)$, after adding T-join edges to G. Hence the total cost of G^* is at least $C_X + C_{IO}(G) + C(G)$. Substituting $C_M + C_U$ for $C_{IO}(G) + C(G)$, we get, $C_M + C_U + C_X \leq C^*$. ∎

5. Modified MIXED algorithm. Figure 2 describes the modified MIXED algorithm. First, we run algorithm INOUTDEGREE on input graph G and obtain $G_{IO} = (V, U, M)$. Before running EVENDEGREE of MIXED1 algorithm, reset the weights of all arcs and edges used by M to 0, forcing EVENDEGREE to duplicate edges/arcs of M whenever possible, as opposed to duplicating edges of U. In other words, use the following cost function when running EVENDEGREE algorithm:

$$c'(e) = \begin{cases} 0 & \text{if } e \in M, \\ c(e) & \text{otherwise.} \end{cases}$$

Use the actual weights for the rest of the MIXED1 algorithm. There are no changes made in MIXED2 algorithm.

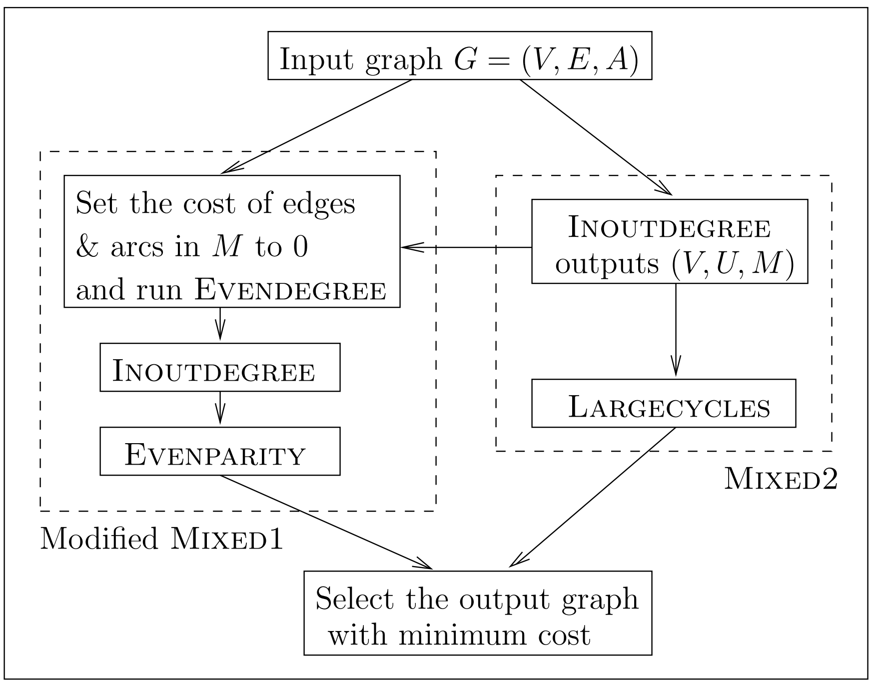


FIG. 2. *Modified* MIXED *algorithm.*

Remark. The cost of arcs and edges of M need not be set to zero. In practice, the cost of each edge in U could be scaled up by a big constant. This ensures that the total cost of edges from U in the minimum-cost T-join is minimized.

5.1. Analysis of Modified MIXED algorithm.

LEMMA 5.1. *Consider the minimum cost T-join of odd-degree supernodes of* UG. *After adding this T-join to* G, *compute a T-join of odd-degree nodes within each equivalence class. These steps together are equivalent to computing the minimum cost T-join of odd-degree nodes of* G *with the new cost function* c'.

Proof. The minimum cost T-join of odd-degree supernodes of UG contains a few edges from U and costs C_X. Once we add these edges to G, each equivalence class of G has an even number of odd-degree nodes. Since the nodes in each equivalence class are connected by edges of M, there exists a T-join of the odd-degree nodes within each equivalence class using only the edges and arcs of M. These two steps together make all nodes of G even degree. Now consider the execution of EVENDEGREE on G with the new cost function c': Since EVENDEGREE is an optimal algorithm, it computes a minimum cost T-join of odd-degree nodes of G. As observed earlier, it induces a T-join of the odd-degree supernodes of UG and therefore costs at least C_X. □

LEMMA 5.2. *Let C_M be the cost of arcs in M and let C^* be the cost of an optimal postman tour of G. The cost of the tour generated by Modified* MIXED1 *algorithm is at most $C^* + C_M$.*

Proof. Consider the components induced by the arcs of M. In the original graph G, the arcs of M correspond to arcs and oriented edges, possibly duplicated. By setting the cost of edges and arcs of M to 0 in G, we force EVENDEGREE not to duplicate the edges of U as much as possible. By Lemmas 4.2 and 5.1, its output graph will contain a few duplicated edges of U (costing C_X) and a few duplicated edges and arcs of M. Let H be the graph at this stage. Note that EVENDEGREE duplicates each arc of M at most once to form H. We follow Frederickson's analysis [8] for the rest of the proof: Let M_1 contain two copies of each arc in M. Clearly, M and M_1 both satisfy INOUT property. Hence, the union of U, X, and M_1 forms an INOUT solution containing H, whose cost is $C_U + C_X + 2 * C_M$. Since INOUTDEGREE is an optimal algorithm for an INOUT problem, it is guaranteed to find an INOUT solution of cost at most $C_U + C_X + 2 * C_M$. This is at most $C^* + C_M$ by Lemma 4.2. Finally, EVENPARITY does not change the cost of the solution. □

LEMMA 5.3 (Frederickson [8]). *Algorithm* MIXED2 *finds a tour whose cost is at most $2C^* - C_M$.*

THEOREM 5.4. *Algorithm Modified* MIXED *produces a tour whose cost is at most $\frac{3}{2}\ C^*$.*

Proof. By Lemma 5.2, Modified MIXED1 outputs a solution whose cost is at most $C_M + C^*$, which is at most $\frac{3}{2}\ C^*$, if $C_M \leq C^*/2$. On the other hand, if $C_M > C^*/2$, then by Lemma 5.3, MIXED2 outputs a solution whose cost is at most $2C^* - C_M$, which is at most $\frac{3}{2}\ C^*$. □

6. Conclusion. We have presented an algorithm and analysis to achieve an approximation ratio of $\frac{3}{2}$ for the mixed postman problem. Improvement in the performance ratio is achieved by proving an improved lower bound on the cost of an optimal postman tour. The performance ratio is tight as shown by Frederickson's examples.

Acknowledgments. We thank an anonymous referee for pointing out an error in an earlier version of our paper. We thank Eva Tardos for suggesting a simpler proof of Lemma 4.1.

REFERENCES

[1] R. K. AHUJA, T. MAGNANTI, AND J. ORLIN, *Network Flows*, Prentice Hall, Englewood Cliffs, NJ, 1993.

[2] P. BRUCKER, *The Chinese postman problem for mixed networks*, in Proceedings of the International Workshop on Graphtheoretic Concepts in Computer Science, Lecture Notes in Comput. Sci. 100, Springer-Verlag, New York, 1980, pp. 354–366.

[3] N. Christofides, E. Benavent, V. Campos, A. Corberan, and E. Mota, *An optimal method for the mixed postman problem*, in System Modelling and Optimization, Lecture Notes in Control and Inform. Sci. 59, Springer, New York, 1984.

[4] J. Edmonds and E. L. Johnson, *Matching, Euler tours and the Chinese postman*, Math. Programming, 5 (1973), pp. 88–124.

[5] H. A. Eiselt, *Arc routing problems, Part* I: *The Chinese postman problem*, Oper. Res., 43 (1995), pp. 231–242.

[6] H. A. Eiselt, *Arc routing problems, Part* II: *The rural postman problem*, Oper. Res., 43 (1995), pp. 399–414.

[7] L. R. Ford and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, NJ, 1962.

[8] G. N. Frederickson, *Approximation algorithms for some postman problems*, J. ACM, 26 (1979), pp. 538–554.

[9] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Co., New York, 1979, p. 212.

[10] M. Guan, *Graphic programming using odd and even points*, Chinese Math., 1 (1962), pp. 273–277.

[11] C. Kappauf and G. Koehler, *The mixed postman problem*, Discrete Appl. Math., 1 (1979), pp. 89–103.

[12] H. Karloff, *Linear Programming*, Birkhauser, Boston, MA, 1991.

[13] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley & Sons, New York, 1988, pp. 648–651.

[14] Y. Nobert and J. C. Picard, *An optimal algorithm for the mixed Chinese postman problem*, Networks, 27 (1996), pp. 95–108.

[15] C. H. Papadimitriou, *On the complexity of edge traversing*, J. ACM, 23 (1976), pp. 544–554.

[16] W. L. Pearn, *Solvable cases of the k-person Chinese postman problem*, Oper. Res. Lett., 16 (1994), pp. 241–244.

[17] W. L. Pearn and C. M. Liu, *Algorithms for the Chinese postman problem on mixed networks*, Comput. & Oper. Res., 22 (1995), pp. 479–489.

[18] T. K. Ralphs, *On the mixed Chinese postman problem*, Oper. Res. Lett., 14 (1993), pp. 123–127.

SIAM J. DISCRETE MATH.
Vol. 12, No. 4, pp. 434–458

ON THE CONTOUR OF RANDOM TREES*

BERNHARD GITTENBERGER†

Abstract. Two stochastic processes describing the contour of simply generated random trees are studied: the contour process as defined by Gutjahr and Pflug [W. Gutjahr and G. Ch. Pflug, *Stochastic Process. Appl.,* 41 (1992), pp. 69–89] and the traverse process constructed of the node heights during pre-order traversal of the tree. Using multivariate generating functions and singularity analysis the weak convergence of the contour process to Brownian excursion is shown and a new proof of the analogous result for the traverse process is obtained.

Key words. random trees, generating functions, singularity analysis, branching processes, Brownian excursion

AMS subject classifications. 05C05, 60J80, 05A16

PII. S0895480195289928

1. Introduction.

Let $\mathcal{A}$ be a class of plane rooted trees and define for $T \in \mathcal{A}$ the size $|T|$ by the number of nodes of T. Furthermore, there is assigned a weight $\omega(T)$ to each $T \in \mathcal{A}$. Let a_n denote the quantity

$$a_n = \sum_{|T|=n} \omega(T).$$

In addition, let us define the generating function (GF) corresponding to $\mathcal{A}$ by $a(z) = \sum_{n\geq 0} a_n z^n$. According to Meir and Moon [11] we call a family of trees *simply generated* if its GF satisfies a functional equation of the form

$$a(z) = z\varphi(a(z)), \tag{1.1}$$

where $\varphi(t) = \sum_{i\geq 0} \varphi_i t^i$ with $\varphi_i \geq 0, \varphi_0 > 0$.

Let $n_k(T)$ denote the number of nodes $v \in T$ with outdegree k (the outdegree of v is the number of edges incident with v that lead away from the root). Then we have for each simply generated tree T the relation

$$\omega(T) = \prod_{k\geq 0} \varphi_k^{n_k(T)}. \tag{1.2}$$

Consider a simply generated tree T of size n. The height $h_T(x)$ of a node $x \in T$ is defined to be the number of edges of the uniquely determined path that connects x with the root. Let $\hat{h}_T(m)$ denote the height of the mth leaf of T, supposing that the leaves are enumerated from left to right. In the following we will assume that for each n the set of all trees of size n is equipped with a probability distribution according to the weights (1.2). Then $\hat{h}_T(m)$ becomes a random variable which we denote by $\hat{H}_n(m)$. If we define the continuation of $\hat{H}_n(m)$ by linear interpolation, i.e.,

$$\hat{H}_n(x) = (\lfloor x \rfloor + 1 - x)\, \hat{H}_n(\lfloor x \rfloor) + (x - \lfloor x \rfloor)\, \hat{H}_n(\lfloor x \rfloor + 1),$$

*Received by the editors August 3, 1995; accepted for publication (in revised form) March 22, 1999; published electronically October 19, 1999. This research was supported by Austrian Science Foundation FWF grant P10187-MAT.

http://www.siam.org/journals/sidma/12-4/28992.html

†Department of Geometry, TU Wien, Wiedner Hauptstrasse 8-10/113, A-1040 Wien, Austria (Bernhard.Gittenberger@tuwien.ac.at).

then we get a continuous stochastic process. The scaled process

$$\hat{X}_n(t) = \frac{1}{\sqrt{n}}\hat{H}_n(tn), \quad 0 \le t \le 1$$

is called the contour process.

We show that for simply generated trees this process converges weakly to Brownian excursion (for the definition and basic properties see [10, p.75]).

THEOREM 1.1. *Let $W^+(t)$ denote Brownian excursion of duration 1. Furthermore, assume that $\varphi(t)$ has a positive or infinite radius of convergence R and $d = ggT\{i|\varphi_i > 0\} = 1$. Moreover, suppose that the equation*

$$t\varphi'(t) = \varphi(t) \tag{1.3}$$

has a minimal positive solution $\tau < R$. Define

$$\sigma^2 = \frac{\tau^2\varphi''(\tau)}{\varphi(\tau)}.$$

Then the contour process $\hat{X}_n(t)$ converges weakly to Brownian excursion, i.e.,

$$\hat{X}_n\left(\frac{\varphi_0}{\varphi(\tau)}t\right) \xrightarrow{w} \frac{2}{\sigma}W^+(t) \tag{1.4}$$

in $C[0,1]$.

For the class of binary trees, (1.4) was established by Gutjahr and Pflug [9], but their method does not seem to be transferable to the general case because it relies on exact enumeration formulae which are available only for binary trees.

Remark 1. The case $d > 1$ can be treated similarly but is technically more involved. The only differences concerning the results are that the limit theorems hold only for $n \equiv 1 mod d$ and that the limiting distribution in local limit theorems has to be multiplied by d. Thus we restrict ourselves to $d = 1$.

Remark 2. Simply generated trees may be considered as trees associated with Galton–Watson branching processes. In this context (1.3) means that the branching process is critical and σ^2 equals the variance of the offspring distribution. Thus, the above theorem also yields a limiting distribution result for branching processes conditioned on the total progeny. For a more detailed discussion of the connection between trees and branching processes see Aldous [2].

In order to define the traverse process we have to use the tree T' defined to be the tree we obtain by attaching T to a single node which serves as the root of T'. Now consider the following traverse procedure:

1. If the current node is v, choose the left-most successor of v that has not yet been traversed (v' is called *successor of* v if it is adjacent to v and $h_T(v') > h_T(v)$). If no such successor exists, return to the previous node.
2. Start at the root and apply step (1) to its successor.

Since in (1) choosing the left-most successor v' is equivalent to choosing the edge (v, v'), each edge is traversed twice and thus the number of steps is $2n$. Let v_i denote the node we arrive at after i steps, and define $h_n(i) = h_T(v_i),\ i = 0, \ldots, 2n$. Assuming again the probability model induced by the weights (1.2), $h_n(i)$ becomes a stochastic process $H_n(i)$ and we continue as above with $H_n(i)$ by linear interpolation. The traverse process is defined by the scaled process

$$X_n(t) = \frac{1}{\sqrt{n}}H_n(2nt), \quad 0 \le t \le 1.$$

The GFs involved in the investigation of $\hat{X}_n(t)$ and $X_n(t)$ are closely related and thus we rather easily obtain the following from (1.4).

THEOREM 1.2. *Under the assumptions of Theorem* 1.1 *the traverse process* $X_n(t)$ *converges weakly to Brownian excursion, i.e.,*

$$X_n(t) \xrightarrow{w} \frac{2}{\sigma} W^+(t)$$

in $C[0,1]$.

This limit theorem was established by Aldous [2] by means of probabilistic techniques (see [1, 3]) and under the slightly weaker condition $\sigma^2 < \infty$. Our approach yields a new proof of this result.

The paper is organized as follows. In section 2 we give a brief description of the basic methods—especially the combinatorial background—used in the following sections. Section 3 is devoted to the proof of Theorem 1.1. Therefore, we have to show the weak convergence of the finite-dimensional distributions and the tightness of the process (see Billingsley [4]). In order to settle the first part of the proof we first consider the three-dimensional distributions, where we prove an invariance property which enables us to essentially simplify the rest of the proof. The last section provides a brief discussion of the traverse process.

2. Basic methods. In order to derive the above-mentioned limit theorems we use the concept of combinatorial constructions introduced by Vitter and Flajolet [12]. Let $\circ$ denote a node and $\mathcal{A}$ a simply generated family of trees. Then every element in $\mathcal{A}$ has the form

$$\{\circ\} \times \mathcal{A} \times \cdots \times \mathcal{A}.$$

Taking into account that we are considering weighted trees, we have to assign the weight φ_i to the above expression if there are i factors $\mathcal{A}$. Thus we get the following symbolic recursion:

$$\mathcal{A} = \varphi_0 \cdot \{\circ\} \,\cup\, \varphi_1 \cdot \{\circ\} \times \mathcal{A} \,\cup\, \varphi_2 \cdot \{\circ\} \times \mathcal{A} \times \mathcal{A} \,\cup\, \cdots.$$

Using the fact that the operations $\cup$ and $\times$ can be translated into sum and product, respectively, of the corresponding GFs, we obtain the functional equation (1.1).

Now let $\theta(T)$ be a characteristic of the tree T in which we are interested. Then we mark the corresponding substructures of T which is equivalent to introducing a new variable in the GF. Thus we get a bivariate GF

$$a(z,u) = \sum_{m,n \ge 0} a_{mn} z^n u^m.$$

The distribution of θ is given by

$$P\{\theta(T) = m : \ |T| = n\} = \frac{a_{mn}}{a_n},$$

where a_{mn} is the coefficient of $z^n u^m$ in $a(z,u)$, denoted by $[z^n u^m] a(z,u)$. We will calculate this distribution by deriving multivariate asymptotic expansions for a_{kn} with uniform error terms. Thus we get a local limit theorem and, due to uniformity, this implies the corresponding weak limit theorem.

Example. $\theta(T)$ equals the number of leaves of T. If a marked node is represented by $\bullet$ and the family of all trees with marked leaves is denoted by $\mathcal{Y}$, then we get the recursion

$$\mathcal{Y} = \varphi_0 \cdot \{\bullet\} \,\cup\, \{\circ\} \times \Phi(\mathcal{Y}),$$

where

$$\Phi(\mathcal{Y}) = \varphi_1 \cdot \{\circ\} \times \mathcal{Y} \,\cup\, \varphi_2 \cdot \{\circ\} \times \mathcal{Y} \times \mathcal{Y} \,\cup\, \cdots.$$

Due to the correspondence

$$\circ \leftrightarrow z,$$
$$\bullet \leftrightarrow uz,$$

translating into GFs gives

$$y(z,u) = \varphi_0 z(u-1) + z\varphi(y(z,u)). \tag{2.1}$$

For further demonstrations of these marking techniques we refer to [7].

To obtain asymptotic expansions we use Cauchy's integral formula combined with singularity analysis following the ideas of Flajolet and Odlyzko [8]. They used the fact that the coefficients of the power series of an analytic function are essentially determined by the behavior of the function near its dominant singularities, i.e., those on the circle of convergence, and they proved the following theorem.

Theorem 2.1 (see [8]). *Let $f(z)$ be analytic in the domain*

$$\Delta = \{z \,|\, |z| \le z_0 + \eta, |arg(z - z_0)| \ge \vartheta\},$$

where $z_0, \eta > 0$, and $0 < \vartheta < \frac{\pi}{2}$. Furthermore, let α be a real number satisfying $\alpha \notin \{0, 1, 2, \ldots\}$. Then

$$f(z) \sim \left(1 - \frac{z}{z_0}\right)^{\alpha} \text{ for } z \to z_0 \text{ in } \Delta \implies [z^n]f(z) \sim \frac{1}{z_0^n n^{\alpha+1}\Gamma(-\alpha)}.$$

Analogous formulae hold for $\mathcal{O}$ and o instead of $\sim$.

Remark. Let $y(z,u)$ be the function defined by (2.1) and $y_{mn} = [u^m z^n]y(z,u)$. Then it can be shown that $\frac{y_{mn}}{a_n}$ satisfy a central limit theorem with mean

$$\frac{1}{a_n}\sum_{m\ge 0} m y_{mn} = \frac{\varphi_0}{\varphi(\tau)} n + \mathcal{O}(1) \tag{2.2}$$

and variance

$$\left(\frac{\varphi_0}{\varphi(\tau)} - \frac{\varphi_0^2}{\varphi(\tau)^2} - \frac{\varphi_0^2 \varphi'(\tau)}{\varphi(\tau)^3 \varphi''(\tau)}\right) n + \mathcal{O}(1), \tag{2.3}$$

where τ is the solution of (1.3) (see [5, 6]). If $L_n(T)$ denotes the number of leaves of a random tree T of size n, then (2.2) and (2.3) imply

$$\lim_{n\to\infty} \frac{L_n(T)}{n} = \frac{\varphi_0}{\varphi(\tau)}, \quad \text{a.s.}$$

Thus the restriction of $\hat{X}_n(t)$ to the interval $[0, \frac{\varphi_0}{\varphi(\tau)}]$ is justified.

3. The contour process.

3.1. Basic functions and their local expansions. Let $\mathcal{A}$ be a family of simply generated trees with GF defined by (1.1) and $m_1 < m_2 < \cdots < m_p$. Consider the set $\mathcal{F}_{k_1 m_1 k_2 m_2 \ldots k_p m_p n} \subseteq \mathcal{A}$ of all trees T with n nodes satisfying $\hat{h}_T(m_i) = k_i$ for $i = 1, \ldots, p$. Set

$$a_{k_1 m_1 \ldots k_p m_p n} = \sum_{T \in \mathcal{F}_{k_1 m_1 \ldots k_p m_p n}} \omega(T),$$

where $\omega(T)$ denote the weight defined by (1.2). Then the finite-dimensional distributions of $H_n(x)$ are given by

$$P\{H_n(m_1) = k_1, \ldots, H_n(m_p) = k_p\} = \frac{a_{k_1 m_1 \ldots k_p m_p n}}{a_n}.$$

Thus we need asymptotic expansions for $a_{k_1 m_1 \ldots k_p m_p n}$ and a_n. When setting up the GFs it turns out that they are composed of three basic functions: obviously, the function $y(z,u)$ defined by (2.1) plays the most important role. The other two functions are composed of $y(z,u)$:

$$\begin{aligned} \phi_1(z,u,v) &= z \sum_{i \geq 1} \varphi_i \sum_{j_1 + j_2 = i-1} y(z,u)^{j_1} y(z,v)^{j_2} \\ &= z \frac{\varphi(y(z,u)) - \varphi(y(z,v))}{y(z,u) - y(z,v)} \end{aligned}$$

and

$$\phi_2(z,u,v,w) = z \sum_{i \geq 2} \varphi_i \sum_{j_1+j_2+j_3 = i-2} y(z,u)^{j_1} y(z,v)^{j_2} y(z,w)^{j_3}.$$

Remark. These functions originate from the following setup: Consider a node to which we attach $i-1$ trees and a marked leaf b. Then leaves of the trees left from b contribute to the number of b while the others do not. Thus, the trees left from b correspond to the GF $y(z,u)$ and the remaining trees to $y(z,1)$. Summing up over all node degrees and keeping in mind that nodes of degree i are weighted by φ_i, we get the GF $zu\phi_1(z,u,1)$. If we replace the marked node by more complicated structures, we will get powers of ϕ_1 or ϕ_2. Of course there may occur functions $\phi_3, \phi_4, \ldots$ (it is obvious how to define them) if we mark more than two leaves, but they prove to be of no importance for the asymptotics in the following.

In order to proceed we need local expansions of these functions near their singularities. We have the following.

LEMMA 3.1 (see [6]). *Let $\varphi(t)$ have a positive or infinite radius of convergence R. Furthermore, assume that $d = ggT\{i|\varphi_i > 0\} = 1$ and that the equation $t\varphi'(t) = \varphi(t)$ has a minimal positive solution $\tau < R$. Then for an $\varepsilon > 0$ there exists a uniquely determined analytic function $z = f(u)$ on $|u-1| < \varepsilon$ such that $f(1) = \frac{1}{\varphi'(\tau)}$ and $y(f(u),u)$ satisfies*

$$\begin{aligned} y &= \varphi_0 z(u-1) + z\varphi(y), \\ 1 &= z\varphi'(y). \end{aligned}$$

$z = f(u)$ is the only singularity of $y(z,u)$ in the domain $|z| \le z_0+\varepsilon$, $\arg(1-\frac{z}{f(u)}) \ne \pi$, where $z_0 = f(1)$. Moreover, inside the domain $\{(z,u) : |z-f(u)| < \varepsilon,\ \arg(1-\frac{z}{f(u)}) \ne \pi\}$ $y(z,u)$ admits the local representation

$$y(z,u) = g(z,u) - h(z,u)\sqrt{1-\frac{z}{f(u)}},$$

where $g(z,u)$ and $h(z,u)$ are analytic functions satisfying

$$g(z_0,1) = \tau \quad \text{and} \quad h(z_0,1) = \sqrt{\frac{2\varphi(\tau)}{\varphi''(\tau)}}.$$

For $d > 1$ analogous representations in the vicinity of $f(u)\exp\left(j\frac{2\pi i}{d}\right)$, $0 \le j < d$ hold.

COROLLARY 1. *Assume $d = 1$. Then $a(z)$ has one and only one singularity $z = z_0$ on the circle of convergence. Furthermore, the local representation*

$$a(z) = \tau - \frac{\tau\sqrt{2}}{\sigma}\sqrt{1-\frac{z}{z_0}} + \mathcal{O}\left(\left|1-\frac{z}{z_0}\right|^2\right)$$

holds near $z = z_0$

COROLLARY 2. *For $a_n = [z^n]a(z)$ we have*

$$a_n = \frac{\tau}{\sigma z_0^n\sqrt{2\pi n^3}}\left(1+\mathcal{O}\left(\frac{1}{n}\right)\right). \tag{3.1}$$

Remark. It is possible to exchange the roles of z and u in the theorem; that means we also have a local representation of the form

$$y(z,u) = \tilde{g}(z,u) - \tilde{h}(z,u)\sqrt{1-\frac{u}{\tilde{f}(z)}}, \tag{3.2}$$

where $\tilde{g}(z,u)$, $\tilde{h}(z,u)$, and $\tilde{f}(z)$ are analytic functions. Using this lemma, local expansions for the above-mentioned basic functions can easily be derived.

LEMMA 3.2 (see [6]). *Set $z = z_0(1+\frac{t}{n})$ and $u_i = 1+\frac{s_i}{m_i}$, $i = 1,2,3$, where $\varepsilon < \frac{m_i}{n}\frac{\varphi(\tau)}{\varphi_0} < 1-\varepsilon$, for arbitrary $\varepsilon > 0$. Furthermore, let $|t| \le \eta n$ and $|s_i| \le \eta m_i$ for sufficiently small $\eta > 0$. Then for $n \to \infty$ the following local expansions hold:*

$$\begin{aligned}
y(z,u_1) - \tau &= -\sqrt{\frac{2\varphi(\tau)}{\varphi''(\tau)}}\sqrt{-\frac{t}{n}-\frac{\varphi_0}{\varphi(\tau)}\frac{s_1}{m_1}} + \mathcal{O}\left(\frac{|s_1|}{m_1}+\frac{|t|}{n}\right),\\
\phi_1(z,u_1,u_2) &= 1 - \frac{\sigma}{\sqrt{2}}\left(\sqrt{-\frac{t}{n}-\frac{\varphi_0}{\varphi(\tau)}\frac{s_1}{m_1}} + \sqrt{-\frac{t}{n}-\frac{\varphi_0}{\varphi(\tau)}\frac{s_2}{m_2}}\right)\\
&\qquad + \mathcal{O}\left(\frac{|t|}{n}+\frac{|s_1|}{m_1}+\frac{|s_2|}{m_2}\right),\\
\phi_2(z,u_1,u_2,u_3) &= \frac{z_0\varphi''(\tau)}{2} + \mathcal{O}\left(\sqrt{\frac{|t|}{n}+\frac{|s_1|}{m_1}+\frac{|s_2|}{m_2}+\frac{|s_3|}{m_3}}\right).
\end{aligned} \tag{3.3}$$

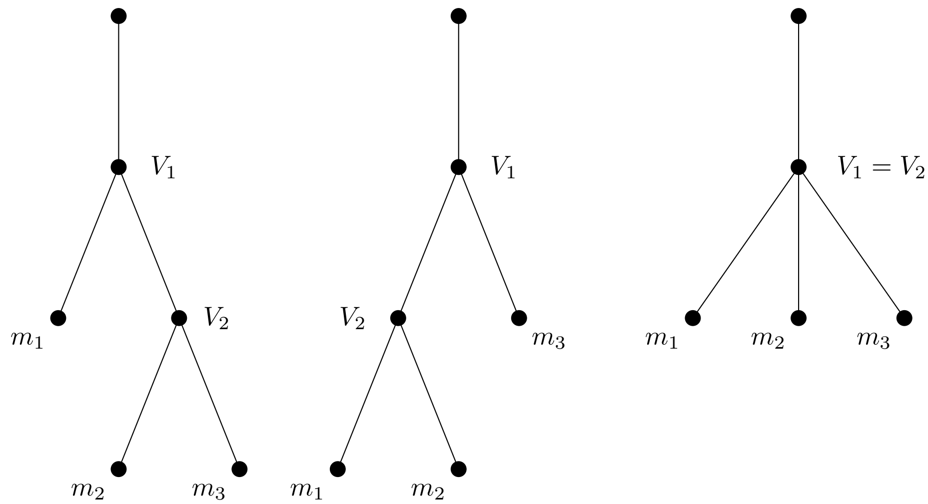


FIG. 3.1. *The possible shapes of T_0 for $p = 3$.*

Since expressions like those in the previous lemma will frequently occur in the following sections, from now on we will use the abbreviation

$$c_{i_1 i_2 \ldots i_k} = \sqrt{-\frac{t}{n} - \frac{\varphi_0}{\varphi(\tau)}\left(\frac{s_{i_1}}{m_{i_1}} + \cdots + \frac{s_{i_k}}{m_{i_k}}\right)},$$

$$c = \sqrt{-\frac{t}{n}}.$$

3.2. An invariance property. Drmota [6] used the above setup to determine the one- and two-dimensional distributions of the contour process. The method works in principle for higher dimensional distributions, too, but the expressions obtained in these cases become too complicated to cope with. If we combine this method with an idea of Gutjahr and Pflug [9] that works for binary trees, we will achieve an essential simplification. The idea is to introduce an additional quantity l_i which is defined as follows: Consider a simply generated tree T, where the leaves with numbers $m_1 < m_2 < \cdots < m_p$ are marked. Then the paths connecting the root with the m_ith leaf and the m_{i+1}st leaf, respectively, have at least the root in common. Let V_i denote that of the common nodes which has maximal height and define $l_i := h_T(V_i)$.

Let us now consider the case $p = 3$. Define the GF

$$B_{k_1 l_1 k_2 l_2 k_3}(z, u_1, u_2, u_3) = \sum_{n, m_1, m_2, m_3 \geq 0} b_{k_1 m_1 l_1 k_2 m_2 l_2 k_3 m_3 n} z^n u_1^{m_1} u_2^{m_2} u_3^{m_3},$$

where $b_{k_1 m_1 l_1 k_2 m_2 l_2 k_3 m_3 n}$ denotes the sum of weights of all trees with n nodes and satisfying $\hat{h}_T(m_i) = k_i, i = 1, 2, 3$, and $h_T(V_j) = l_j$. To set up this GF we have to distinguish three cases: $l_1 < l_2$, $l_1 > l_2$, and $l_1 = l_2$. The third one is asymptotically negligible since it corresponds to a hyperplane in $\mathbf{R}^5$ in the limit case and thus it has no influence on the density of the limiting distribution (for a detailed argumentation see [9]).

For convenience, we introduce a tree T_0 consisting of m_1, m_2, m_3, V_1, V_2, and the root of T. The edges of T_0 are the paths that connect its nodes in T (see Figure 3.1). Now consider a node x of T which lies on the edge of T_0 which connects the root with V_1. As mentioned in the previous section, the leaves of all trees which are rooted in x

and lying left from the path containing x contribute to the number of m_1, m_2, and m_3 while leaves of those trees lying on the right-hand side yield no contribution. Thus, the subgraph of T induced by x and all its descending trees not lying in T_0 corresponds to the GF $\phi_1(z, u_1u_2u_3, 1)$. If x lies on a different path, we have to observe which of the leaves m_1, m_2, and m_3 are to the left or right of the trees rooted in x. For instance, if $x \in (V_1, V_2)$, then the corresponding GF is $\phi_1(z, u_2u_3, 1)$ (where we assumed T_0 to be the left-most tree in Figure 3.1). Thus each edge of T_0 corresponds to a power of ϕ_1 according to its length and with suitably chosen arguments. The branching points V_1 and V_2 yield factors ϕ_2 due to the fact that we have to distinguish three classes of trees rooted at V_1 or V_2: the ones to the left of all edges of T_0, the ones to the right of those edges, and the ones lying in between. This yields, e.g., for V_2 the GF $\phi_2(z, u_2u_3, u_3, 1)$. Finally, we have to take into account the leaves m_1, m_2, and m_3, yielding the GFs $\varphi_0 z u_1u_2u_3$, $\varphi_0 z u_2u_3$, and $\varphi_0 z u_3$, respectively. Therefore, we obtain for $l_1 < l_2$

$$\begin{aligned} B_{k_1l_1k_2l_2k_3}(z, u_1, u_2, u_3) = \varphi_0^3 z^3 u_1 u_2^2 u_3^3 \phi_2(z, u_1u_2u_3, u_2u_3, 1)\phi_2(z, u_2u_3, u_3, 1) \\ \times\phi_1(z, u_1u_2u_3, 1)^{l_1}\phi_1(z, u_1u_2u_3, u_2u_3)^{k_1-l_1-1} \\ \times\phi_1(z, u_2u_3, 1)^{l_2-l_1-1}\phi_1(z, u_2u_3, u_3)^{k_2-l_2-1} \\ \times\phi_1(z, u_3, 1)^{k_3-l_2-1}. \end{aligned}$$

Analogously, we get the GF in the case $l_1 > l_2$:

$$\begin{aligned} B_{k_1l_1k_2l_2k_3}(z, u_1, u_2, u_3) = \varphi_0^3 z^3 u_1 u_2^2 u_3^3 \phi_2(z, u_1u_2u_3, u_3, 1)\phi_2(z, u_1u_2u_3, u_2u_3, u_3) \\ \times\phi_1(z, u_1u_2u_3, 1)^{l_2}\phi_1(z, u_1u_2u_3, u_2u_3)^{k_1-l_1-1} \\ \times\phi_1(z, u_1u_2u_3, u_3)^{l_1-l_2-1}\phi_1(z, u_2u_3, u_3)^{k_2-l_1-1} \\ \times\phi_1(z, u_3, 1)^{k_3-l_2-1}. \end{aligned}$$

If we consider random trees, then the heights of the leaves m_1, m_2, m_3 as well as the path lengths l_1, l_2 become random variables. Let us denote this multivariate random variable by $(K_1, L_1, K_2, L_2, K_3)$. Its distribution is determined by

$$[z^n u_1^{m_1} u_2^{m_2} u_3^{m_3}] B_{k_1l_1k_2l_2k_3}(z, u_1, u_2, u_3).$$

This coefficient can be calculated asymptotically by means of Cauchy's integral formula. The integration path is chosen in such a way that one part lies close to the singularity (this part yields the main term) and the remaining part is asymptotically negligible (for details see the next section). Thus, the limiting distribution is completely determined by the local behavior of the GF.

Let k_i, $i = 1, 2, 3$, and l_j, $j = 1, 2$, be proportional to $\sqrt{n}$ and m_i, $i = 1, 2, 3$, satisfy the condition $\varepsilon < \frac{m_i}{n}\frac{\varphi(\tau)}{\varphi_0} < 1 - \varepsilon$ for arbitrary $\varepsilon > 0$. Using Lemma 3.2 and the fact that $k_i - 1 \sim k_i$ and $l_i - 1 \sim l_i$, it can be shown that $B_{k_1l_1k_2l_2k_3}(z, u_1, u_2, u_3)$ admits the local representation

$$\begin{aligned} \frac{\varphi_0^3 z_0^5 \varphi''(\tau)^2}{4} \exp\Bigg(-\frac{\sigma}{\sqrt{2}}(l_1(c_{123} + c) + (k_1 - l_1)(c_{123} + c_{23}) + (l_2 - l_1)(c_{23} + c) \\ +(k_2 - l_2)(c_{23} + c_3) + (k_3 - l_2)(c_3 + c))\Bigg) \qquad (3.4) \\ \times\left(1 + \mathcal{O}\left(M\left(\frac{|t|}{n} + \frac{|s_1|}{m_1} + \frac{|s_2|}{m_2} + \frac{|s_3|}{m_3}\right)\right) + \mathcal{O}\left(\sqrt{\frac{|t|}{n} + \frac{|s_1|}{m_1} + \frac{|s_2|}{m_2} + \frac{|s_3|}{m_3}}\right)\right) \end{aligned}$$

for $l_1 < l_2$ and

$$\frac{\varphi_0^3 z_0^5 \varphi''(\tau)^2}{4} \exp\left(-\frac{\sigma}{\sqrt{2}}(l_2(c_{123}+c)+(k_1-l_1)(c_{123}+c_{23})+(l_1-l_2)(c_{123}+c_3)\right.$$
$$\left.+(k_2-l_1)(c_{23}+c_3)+(k_3-l_2)(c_3+c))\right)$$
$$\times\left(1+\mathcal{O}\left(M\left(\frac{|t|}{n}+\frac{|s_1|}{m_1}+\frac{|s_2|}{m_2}+\frac{|s_3|}{m_3}\right)\right)+\mathcal{O}\left(\sqrt{\frac{|t|}{n}+\frac{|s_1|}{m_1}+\frac{|s_2|}{m_2}+\frac{|s_3|}{m_3}}\right)\right)$$

for $l_1 > l_2$, respectively, where $M = \max(k_1, k_2, k_3)$. The difference of the exponents is

$$(l_1 - l_2)(c_{123} + c - c_{23} - c - c_{123} - c_3 + c_{23} + c_3) = 0$$

and thus (3.4) also holds for the case $l_1 > l_2$, meaning that the local representation of $B_{k_1 l_1 k_2 l_2 k_3}(z, u_1, u_2, u_3)$ is invariant with respect to the shape of T_0. Generalizing the above considerations we get the following.

LEMMA 3.3 (invariance property). *Let $B_{k_1 l_1 \dots k_{p-1} l_{p-1} k_p}(z, u_1, \dots, u_p)$ be the GF of $(K_1, L_1, \dots, K_{p-1}, L_{p-1}, K_p)$ and $\mathcal{B}_p$ denote the set of all binary trees with p leaves, where p is a fixed positive integer. Assume that $T_0 \in \mathcal{B}_p$ and that the quantities $k_i^2, m_i, i = 1, \dots, p$, and $l_j^2, j = 1, \dots, p-1$, are asymptotically proportional to n. Then for $\|z - z_0, u_1 - 1, \dots, u_p - 1\|_{\max} = o(\sqrt{n})$ there exists a local asymptotic representation of $B_{k_1 l_1 \dots k_{p-1} l_{p-1} k_p}(z, u_1 \dots, u_p)$ that holds for all $T_0 \in \mathcal{B}_p$.*

Proof. As the lemma is intended to simplify the proofs in the following section, we have to consider the one special shape of T_0 which is most convenient to work with and then show that the local representation is invariant with respect to the shape of T_0. Thus, we choose the one that satisfies $l_1 < l_2 < \dots < l_p$ (according to Figure 3.2) in order to get rid of the usually unpleasant terms $\min(l_i, l_j)$ and $\max(l_i, l_j)$ occurring in the GFs. This leads to the GF

$$B(z, u_1, \dots, u_p) = \varphi_0^p z^p u_1 u_2^2 \cdots u_p^p \prod_{i=1}^{p-1} \left[\phi_1(z, u_i \cdots u_p, u_{i+1} \cdots u_p)^{k_i - l_i - 1}\right.$$
$$\left.\times \phi_1(z, u_i \cdots u_p, 1)^{l_i - l_{i-1} - 1 + \delta_{i1}} \phi_2(z, u_i \cdots u_p, u_{i+1} \cdots u_p)\right]$$
$$\times \phi_1(z, u_p, 1)^{k_p - l_{p-1} - 1},$$

where we define $l_0 = 0$ and δ_{ij} is the Kronecker delta defined by $\delta_{ij} = 1 - \operatorname{sgn}|i - j|$. Let $z = z_0\left(1 + \frac{t}{n}\right)$ and $u_i = 1 + \frac{s_i}{m_i}$ be chosen in such a way that the assumptions of Lemma 3.2 hold. Then we get for $k_i = \kappa_i \sqrt{n}$ and $l_i = \lambda_i \sqrt{n}$ the local representation

$$B(z, u_1, \dots, u_p) = \varphi_0^p z_0^p \left(\frac{z_0 \varphi''(\tau)}{2}\right)^{p-1} \exp\left(-\frac{\sigma}{\sqrt{2}}\left(\sum_{i=1}^{p-1}((k_i - l_i)(c_{i \cdots p} + c_{i+1, \dots, p})\right.\right.$$
$$\left.\left.+(l_i - l_{i-1})(c_{i \cdots p} + c)) + (k_p - l_{p-1})(c_p + c)\right)\right)$$
$$\times\left(1+\mathcal{O}\left(M_p\left(\frac{|t|}{n}+\sum_{i=1}^{p}\frac{|s_i|}{m_i}\right)\right)+\mathcal{O}\left(\sqrt{\frac{|t|}{n}+\sum_{i=1}^{p}\frac{|s_i|}{m_i}}\right)\right), \tag{3.5}$$

where $M_p = \max_{1 \le i \le p} k_i$.

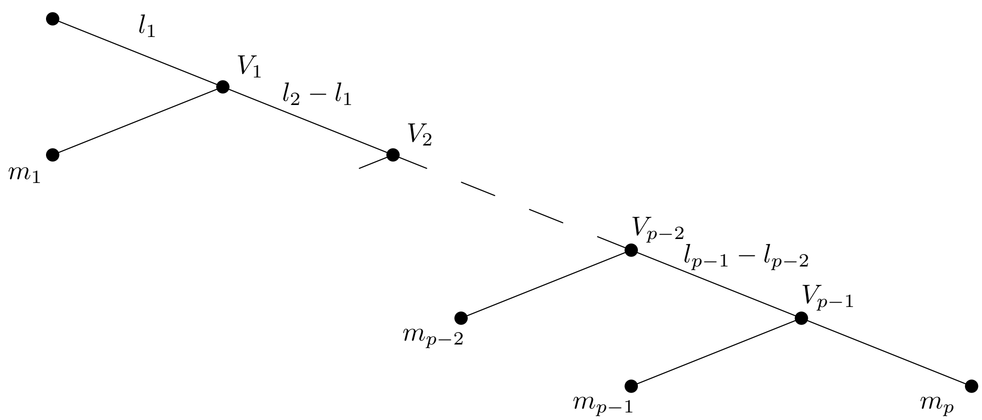


FIG. 3.2. *The chosen shape of* T_0.

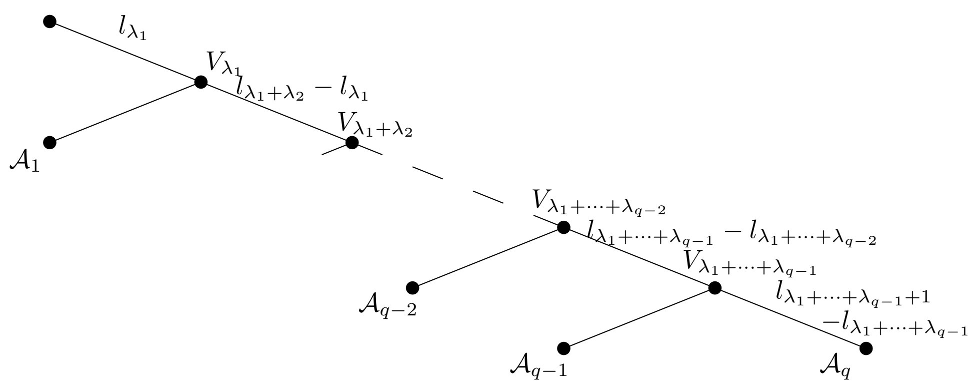


FIG. 3.3. *The general shape of* T_0.

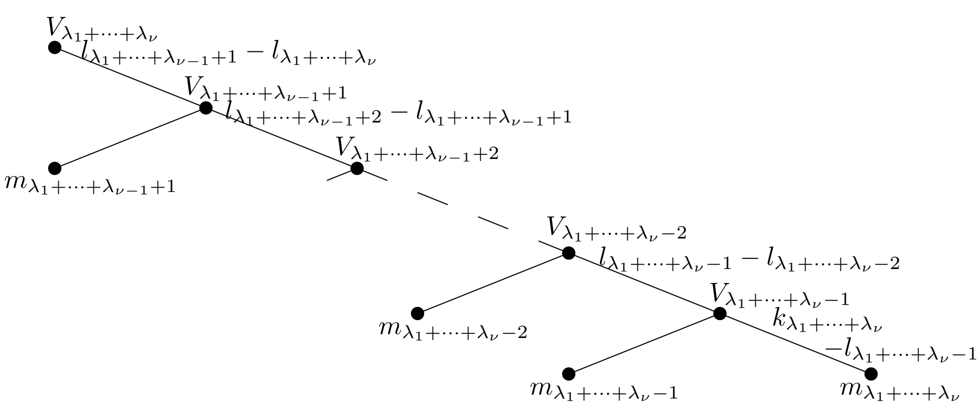


FIG. 3.4. *Zooming into* $\mathcal{A}_\nu$.

From this formula it is easy to see how the structure of T_0 can be translated into the proper terms of the exponent in the local expansion of the corresponding GF. We complete our proof by induction on p. Note that the general shape of T_0 has the form of Figure 3.3, where $\mathcal{A}_\nu$ are trees with λ_ν marked leaves such that $\sum_{\nu=1}^{q} \lambda_\nu = p$. The GF corresponding to $\mathcal{A}_\nu$ is a product of ϕ_1- and ϕ_2-terms, and by the induction hypothesis the local behavior is independent of the shape of the underlying part of T_0 which we denote by $T_{0\nu}$. Thus, we may assume that $T_{0\nu}$ is already "well shaped," i.e., as shown in Figure 3.4. By (3.5) the exponent in the expansion of the GF corresponding to $\mathcal{A}_\nu \cup (V_{\lambda_1+\cdots+\lambda_\nu}, V_{\lambda_1+\cdots+\lambda_{\nu-1}+1})$ (where (i,j) denotes the edge in T_0 which connects i and j) is given by

$$\begin{aligned}\sum_{i=\Lambda_{\nu-1}+1}^{\Lambda_\nu} &[(k_i-l_i)(c_{i\cdots p}+c_{i+1,\ldots,p}) \\ &\quad + (l_i - l_{i-1} + \delta_{i,\Lambda_{\nu-1}+1}(l_{i-1}-l_{\Lambda_\nu})(c_{i\cdots p}+c_{\Lambda_\nu+1,\ldots,p})] \\ &\quad + (k_{\Lambda_\nu}-l_{\Lambda_\nu-1})\left(c_{\Lambda_\nu,\ldots,p}+c_{\Lambda_\nu+1,\ldots,p}\right),\end{aligned}$$

where we defined, for convenience, $\Lambda_\nu = \lambda_1 + \cdots + \lambda_\nu$, and furthermore we have to set $\Lambda_0 := 0$, $l_{\Lambda_q} := l_{\Lambda_{q-1}}$, and $c_{\Lambda_q+1,\ldots,p} := c$. Hence we have to show the following identity:

$$\begin{aligned}&\sum_{\nu=1}^{q}\Bigg[\sum_{i=\Lambda_{\nu-1}+1}^{\Lambda_\nu-1}[(k_i-l_i)(c_{i\cdots p}+c_{i+1,\ldots,p}) \\ &\quad + (l_i-l_{i-1}+\delta_{i,\Lambda_{\nu-1}+1}(l_{i-1}-l_{\Lambda_\nu}))(c_{i\cdots p}+c_{\Lambda_\nu+1,\ldots,p})] \\ &\quad + (k_{\Lambda_\nu}-l_{\Lambda_\nu-1})\left(c_{\Lambda_\nu,\ldots,p}+c_{\Lambda_\nu+1,\ldots,p}\right)\Bigg] + \sum_{\nu=1}^{q-1}(l_{\Lambda_\nu}-l_{\Lambda_\nu-1})\left(c_{\Lambda_{\nu-1}+1,\ldots,p}+c\right) \\ &= \sum_{i=1}^{p-1}[(k_i-l_i)(c_{i\cdots p}+c_{i+1,\ldots,p})+(l_i-l_{i-1})(c_{i\cdots p}+c)]+(k_p-l_{p-1})(c_p+c).\end{aligned}$$

Subtracting $\sum_{i=1}^{p-1}[(k_i-l_i)(c_{i\cdots p}+c_{i+1,\ldots,p})+(l_i-l_{i-1})c_{i\cdots p}]$ gives

$$\begin{aligned}&\sum_{\nu=1}^{q}\Bigg[\sum_{i=\Lambda_{\nu-1}+1}^{\Lambda_\nu-1}(l_i-l_{i-1}+\delta_{i,\Lambda_{\nu-1}+1}(l_{i-1}-l_{\Lambda_\nu}))c_{\Lambda_\nu+1,\ldots,p}+\left(l_{\Lambda_{\nu-1}}-l_{\Lambda_\nu}\right)c_{\Lambda_{\nu-1}+1,\ldots,p} \\ &\quad + (k_{\Lambda_\nu}-l_{\Lambda_\nu-1})\left(c_{\Lambda_\nu,\ldots,p}+c_{\Lambda_\nu+1,\ldots,p}\right)\Bigg]+\sum_{\nu=1}^{q-1}(l_{\Lambda_\nu}-l_{\Lambda_\nu-1})\left(c_{\Lambda_{\nu-1}+1,\ldots,p}+c\right) \\ &\quad -\sum_{\nu=1}^{q-1}[(k_{\Lambda_\nu}-l_{\Lambda_\nu})\left(c_{\Lambda_\nu,\ldots,p}+c_{\Lambda_\nu+1,\ldots,p}\right)+(l_{\Lambda_\nu}-l_{\Lambda_\nu-1})\,c_{\Lambda_\nu,\ldots,p}] \\ &= k_p(c_p+c)-l_{p-1}c_p,\end{aligned}$$

and this can be easily checked. ∎

3.3. The finite-dimensional distributions. Applying the substitution $w_i = k_i + k_{i+1} - 2l_i$ on (3.5) yields

$$B(z,u_1,\ldots,u_p)=\varphi_0^p z_0^p\left(\frac{z_0\varphi''(\tau)}{2}\right)^{p-1}\exp\left(-\frac{\sigma}{\sqrt{2}}\left(c_{1\cdots p}k_1+\sum_{i=1}^{p-1}c_{i+1,\ldots,p}w_i+ck_p\right)\right)$$
$$\times\left(1+\mathcal{O}\left(M_p\left(\frac{|t|}{n}+\sum_{i=1}^{p}\frac{|s_i|}{m_i}\right)\right)\right), \tag{3.6}$$

and by means of this formula we are able to prove the following.

THEOREM 3.1. *Let $\varepsilon>0$ and $w_i=k_i+k_{i+1}-2l_i$. Then we have, uniformly for $\frac{m_1}{n}\ge\varepsilon$, $\frac{m_{j+1}-m_j}{n}\ge\varepsilon$, $j=1,\ldots,p-1$, $\frac{\varphi_0}{\varphi(\tau)}-\frac{m_p}{n}\le\varepsilon$, and $k_i=\mathcal{O}(\sqrt{n})$, $i=1,\ldots,p$, $w_j=\mathcal{O}(\sqrt{n})$, $j=1,\ldots,p-1$,*

$$[z^n u_1^{m_1}\cdots u_p^{m_p}]B=C_p' k_1 w_1\cdots w_{p-1}k_p$$
$$\times\left[m_1(m_2-m_1)\cdots(m_p-m_{p-1})\left(n-\frac{\varphi(\tau)}{\varphi_0}m_p\right)\right]^{-3/2}$$
$$\times\exp\left(-\frac{\sigma^2}{8}\frac{\varphi_0}{\varphi(\tau)}\left(\frac{k_1^2}{m_1}+\sum_{i=2}^{p}\frac{w_{i-1}^2}{m_i-m_{i-1}}+\frac{k_p^2}{\frac{\varphi_0}{\varphi(\tau)}n-m_p}\right)\right)$$
$$\times\left(1+\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)\right),\quad n\to\infty, \tag{3.7}$$

where

$$C_p'=\varphi_0^p z_0^{p-n}\left(\frac{\sigma}{\sqrt{2}}\right)^{p+1}\left(\frac{z_0\varphi''(\tau)}{2}\right)^{p-1}\frac{1}{(2\sqrt{\pi})^{p+1}}\left(\frac{\varphi_0}{\varphi(\tau)}\right)^{p/2}$$
$$=z_0^{-n}\frac{\tau}{2^{p+1}}\left(\frac{\sigma}{\sqrt{2}}\right)^{3p-1}\pi^{-(p+1)/2}\left(\frac{\varphi_0}{\varphi(\tau)}\right)^{3p/2}.$$

Dividing (3.7) by a_n yields the following local limit theorem.

COROLLARY. *Let $k_j=\kappa_j\sqrt{n}+o(\sqrt{n})\in\mathbf{N}$, $j=1,\ldots,p$, and $w_j=k_i+k_{i+1}-2l_i=\omega_j\sqrt{n}+o(\sqrt{n})\in\mathbf{N}$, $j=1,\ldots,p-1$, satisfying $|\kappa_{j+1}-\kappa_j|\le\omega_j\le\kappa_{j+1}+\kappa_j$. Moreover, assume $\frac{\varphi(\tau)}{\varphi_0}m_j=\mu_j n+o(n)$, where $0<\mu_1<\cdots<\mu_p<1$, and let W_i denote the random variable $K_i+K_{i+1}-2L_i$. Then the density*

$$P\{K_1=k_1,W_1=w_1,\ldots,K_{p-1}=k_{p-1},W_{p-1}=w_{p-1},K_p=k_p\}$$
$$=\frac{b_{k_1m_1l_1\cdots k_{p-1}m_{p-1}l_{p-1}k_pm_pn}}{a_n}$$

of the random variable $(K_1,W_1,\ldots,K_{p-1},W_{p-1},K_p)$ admits the following asymptotic expansion:

$$n^{(2p-1)/2}\frac{b_{k_1m_1l_1\cdots k_{p-1}m_{p-1}l_{p-1}k_pm_pn}}{a_n}=\frac{1}{(2\sqrt{\pi})^p}\left(\frac{\sigma}{\sqrt{2}}\right)^{3p}\kappa_1\omega_1\cdots\omega_{p-1}\kappa_p$$
$$\times\left[\mu_1(\mu_2-\mu_1)\cdots(\mu_p-\mu_{p-1})(1-\mu_p)\right]^{-3/2}$$
$$\times\exp\left(-\frac{\sigma^2}{8}\left(\frac{\kappa_1^2}{\mu_1}+\sum_{j=2}^{p}\frac{\omega_{j-1}^2}{\mu_j-\mu_{j-1}}+\frac{\kappa_p^2}{1-\mu_p}\right)\right)$$
$$+o(1) \tag{3.8}$$

for $n \to \infty$. The error term is uniform in ω_i, $i = 1, \dots, p-1$, and for $\kappa_j \in [a_j, b_j]$, $b_j > a_j > 0$ and $\kappa_{j+1} - \kappa_j > \varepsilon > 0$, $j = 1, \dots, p$.

Now the finite-dimensional distribution of the contour process, i.e., the distribution of $(K_1, \dots, K_p)$, can be calculated. Due to uniformity of the error term it suffices to determine the marginal density in $(\kappa_1, \dots, \kappa_p)$ of (3.8). Doing this we obtain a multivariate Maxwell distribution which actually coincides with that of Brownian excursion. Thus the following theorem holds.

THEOREM 3.2. *Let $\pi_{t_1,\dots,t_k}$ be the projection defined by*

$$\pi_{t_1,\dots,t_k}\colon \quad C[0,1] \to \mathbf{R}^k, \qquad x(t) \mapsto (x(t_1), \dots, x(t_k)).$$

Then the following limit theorem holds:

$$\pi_{t_1,\dots,t_k}\left(\hat{X}_n\left(\frac{\varphi_0}{\varphi(\tau)}t\right)\right) \xrightarrow{d} \pi_{t_1,\dots,t_k}\left(\frac{2}{\sigma}W^+(t)\right).$$

Remark. Note that Theorem 3.1 and its corollary provide only the distributions at the vertices of the polygon $\hat{X}_n(t)$. Thus they imply a slightly different form of the above limit theorem: we have to substitute $\hat{X}_n(t)$ with the corresponding step function process $\hat{X}_n(\lfloor t_n \rfloor / n)$. However, by means of the proof of tightness (see section 3.6) we are able to prove the theorem as we stated it (see the end of section 3).

3.4. Proof of Theorem 3.1. Determination of the main term. In order to prove Theorem 3.1 we use Cauchy's integral formula

$$[z^n u_1^{m_1} \cdots u_p^{m_p}] B(z, u_1, \dots, u_p) = \frac{1}{(2\pi i)^{p+1}} \int_{\Gamma_1} \cdots \int_{\Gamma_p} \int_{\Gamma_0} \frac{B(z, u_1, \dots, u_p)}{z^{n+1} u_1^{m_1+1} \cdots u_p^{m_p+1}} \, dz \, du_p \cdots du_1 \tag{3.9}$$

with the following integration contour: Let z run through the contour $\Gamma_0 = \Gamma_{01} \cup \Gamma_{02} \cup \Gamma_{03} \cup \Gamma_{04}$ defined by

$$\Gamma_{01} = \left\{ z = z_0\left(1 + \frac{t}{n}\right) \middle| \Re t \le 0 \text{ and } |t| = 1 \right\},$$

$$\Gamma_{02} = \left\{ z = z_0\left(1 + \frac{t}{n}\right) \middle| \Im t = 1 \text{ and } 0 \le \Re t \le \log^2 n \right\},$$

$$\Gamma_{03} = \overline{\Gamma}_{02},$$

$$\Gamma_{04} = \left\{ z \,\middle|\, |z| = z_0 \left|1 + \frac{\log^2 n + i}{n}\right| \text{ and } \arg\left(1 + \frac{\log^2 n + i}{n}\right) \le |\arg(z)| \le \pi \right\}.$$

Note that while z is running through Γ_0 the location of the singularity also changes. This fact has to be taken care of when choosing the integration contour for the other variables. The location of the singularity is determined asymptotically by the equations

$$\begin{aligned}
\frac{t}{n} &= 0,\\
\frac{\varphi_0}{\varphi(\tau)}\frac{s_p}{m_p} + \frac{t}{n} &= 0,\\
\frac{\varphi_0}{\varphi(\tau)}\left(\frac{s_{p-1}}{m_{p-1}} + \frac{s_p}{m_p}\right) + \frac{t}{n} &= 0,\\
&\vdots\\
\frac{\varphi_0}{\varphi(\tau)}\left(\frac{s_1}{m_1} + \cdots + \frac{s_{p-1}}{m_{p-1}} + \frac{s_p}{m_p}\right) + \frac{t}{n} &= 0,
\end{aligned}$$

as one can easily see by looking at (3.6). Thus, as the integration contour of $u_1, \ldots, u_p$ we may choose $\Gamma_j = \Gamma_{j1} \cup \Gamma_{j2} \cup \Gamma_{j3} \cup \Gamma_{j4}$ defined by

$$\begin{aligned}
\Gamma_{j1} = \Bigg\{ u_j = \left(1 + \frac{s_j}{m_j}\right) \Bigg| \, &\Re s_j \le -R_j(s_{j+1}, \ldots, s_p, t) \text{ and}\\
&|s_j + R_j(s_{j+1}, \ldots, s_p, t) + I_j(s_{j+1}, \ldots, s_j, t)i| = 1 \Bigg\},\\
\Gamma_{j2} = \Bigg\{ u_j = \left(1 + \frac{s_j}{m_j}\right) \Bigg| \, &\Im s_j = -I_j(s_{j+1}, \ldots, s_p, t) + 1,\\
-R_j(s_{j+1}, \ldots, s_p, t) \le \Re s_j &\text{ and } |u_j| \le \left|1 + \frac{\log^2 m_j + i}{m_j}\right| \Bigg\},\\
\Gamma_{j3} = \Bigg\{ u_j = \left(1 + \frac{s_j}{m_j}\right) \Bigg| \, &\Im s_j = -I_j(s_{j+1}, \ldots, s_p, t) - 1,\\
-R_j(s_{j+1}, \ldots, s_p, t) \le \Re s_j &\text{ and } |u_j| \le \left|1 + \frac{\log^2 m_j + i}{m_j}\right| \Bigg\},
\end{aligned}$$

$$\Gamma_{j4} = \left\{ u_j \, \Bigg| \, |u_j| = \left|1 + \frac{\log^2 m_j + i}{m_j}\right| \text{ and } \arg u_j \in [-\pi, \arg z_{j3}] \cup [\arg z_{j2}, \pi] \right\},$$

where

$$R_j(s_{j+1}, \ldots, s_p, t) = \begin{cases} \max\left(0, \dfrac{\varphi(\tau)}{\varphi_0}\dfrac{m_p}{n}\Re t\right) & \text{if } j = p,\\ \max\left(0, \Re\left(\dfrac{\varphi(\tau)}{\varphi_0}\dfrac{t m_j}{n} + \dfrac{s_{j+1} m_j}{m_{j+1}} + \cdots + \dfrac{s_p m_j}{m_p}\right)\right) & \text{else,} \end{cases}$$

$$I_j(s_{j+1}, \ldots, s_p, t) = \begin{cases} \max\left(n^{2/3}, \dfrac{\varphi(\tau)}{\varphi_0}\dfrac{m_p}{n}\Im t\right) & \text{if } j = p,\\ \max\left(n^{2/3}, \Im\left(\dfrac{\varphi(\tau)}{\varphi_0}\dfrac{t m_j}{n} + \dfrac{s_{j+1} m_j}{m_{j+1}} + \cdots + \dfrac{s_p m_j}{m_p}\right)\right) & \text{else,} \end{cases}$$

and z_{jk} denotes the point of Γ_{jk} with maximal absolute value.

Remark. The functions R_j and I_j guarantee that the Hankel-like contours[1] $\Gamma'_j = \Gamma_{j1} \cup \Gamma_{j2} \cup \Gamma_{j3}$ follow the movement of the singularity while $z, u_{j+1}, \ldots, u_p$ are varying. It can be shown that, for these variables moving away from the Hankel contour along $\Gamma_{.4}$, the singularity drifts out of the circle determined by Γ_{j4} and reaches a point x with $|x| = 1 + Cn^{-1/3}$ when one of the variables $z, u_{j+1}, \ldots, u_p$ arrives at distance $n^{-1/3}$ from the Hankel contour. Thus the term $n^{2/3}$ in the definition of I_j is justified.

Let us now consider the contribution of the Hankel integrals which yields the main term, as we will show in the next section. If we apply the substitutions $z = z_0\left(1 + \frac{t}{n}\right)$, $u_j = 1 + \frac{s_j}{m_j}$ to (3.9) and use the asymptotic expansion (3.6), then we get

$$\frac{C_p}{(2\pi i)^{p+1}} \int_{\Gamma'_0}\int_{\Gamma'_1}\cdots\int_{\Gamma'_p} \exp\left(-\frac{\sigma}{\sqrt{2}}\left(c_{1\cdots p}k_1 + \sum_{i=1}^{p-1} c_{i+1,\ldots,p}w_i + ck_p\right)\right.$$
$$\left. - t - s_1 - \cdots - s_p\right)\frac{ds_p}{m_p}\cdots\frac{ds_1}{m_1}\frac{dt}{n}\left(1 + \mathcal{O}\left(M_p\left(\frac{1}{n} + \sum_{j=1}^{p}\frac{1}{m_j}\right)\right)\right),$$

where

$$C_p = \varphi_0^p z_0^{p-n}\left(\frac{z_0\varphi''(\tau)}{2}\right)^{p-1}.$$

The shape of this integral suggests the substitution

$$\begin{pmatrix} \frac{\varphi_0}{\varphi(\tau)m_1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & & \frac{\varphi_0}{\varphi(\tau)m_p} & 0 \\ 0 & & 0 & \frac{1}{n} \end{pmatrix}\begin{pmatrix} v_1 \\ \vdots \\ v_p \\ t \end{pmatrix}$$
$$= \begin{pmatrix} \frac{\varphi_0}{\varphi(\tau)m_1} & \frac{\varphi_0}{\varphi(\tau)m_2} & \cdots & \frac{\varphi_0}{\varphi(\tau)m_p} & \frac{1}{n} \\ 0 & \frac{\varphi_0}{\varphi(\tau)m_2} & \cdots & \frac{\varphi_0}{\varphi(\tau)m_p} & \frac{1}{n} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & & \ddots & \frac{\varphi_0}{\varphi(\tau)m_p} & \frac{1}{n} \\ 0 & \cdots & \cdots & 0 & \frac{1}{n} \end{pmatrix}\begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_p \\ t \end{pmatrix}$$

which finally leads to

(3.10)
$$\frac{C_p}{m_1\cdots m_p n}\prod_{j=1}^{p}\int_{\gamma_j} e^{-\alpha_j\sqrt{-v_j}-\beta_j v_j}\,dv_j \int_{\gamma_0} e^{-\alpha_{p+1}\sqrt{-t}-\beta_{p+1}t}\,dt\left(1 + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)\right),$$

[1] According to Hankel's representation of the gamma function we will refer to the integration contour starting at $e^{2\pi i}\infty$, passing the origin clockwise and returning to $+\infty$ as Hankel contour. Similarly, we will use the attribute Hankel for all related concepts such as Hankel integral, etc.

where

$$\alpha_1 = \frac{\sigma k_1}{\sqrt{2m_1}}\sqrt{\frac{\varphi_0}{\varphi(\tau)}}, \qquad \beta_1 = 1,$$

$$\alpha_j = \frac{\sigma w_{j-1}}{\sqrt{2m_j}}\sqrt{\frac{\varphi_0}{\varphi(\tau)}}, \quad j=2,\dots,p, \qquad \beta_j = 1 - \frac{m_{j-1}}{m_j}, \quad j=2,\dots,p,$$

$$\alpha_{p+1} = \frac{\sigma k_p}{\sqrt{2n}}, \qquad \beta_{p+1}, \qquad 1-\frac{m_p}{n}\frac{\varphi(\tau)}{\varphi_0},$$

and γ_j are Hankel contours meeting the constraint

$$\Re t \le \log^2 n \quad \text{and} \quad \Re v_j \le \log^2 m_j,\ j=1,\dots,p.$$

LEMMA 3.4. *Let γ be a Hankel contour truncated at K. Then we have for $\alpha,\ \beta > 0$*

$$\frac{1}{2\pi i}\int_\gamma e^{-\alpha\sqrt{-t}-\beta t}\,dt = \frac{\alpha\beta^{-\frac{3}{2}}}{2\sqrt{\pi}}\exp\left(-\frac{\alpha^2}{4\beta}\right) + \mathcal{O}\left(\frac{1}{\beta}e^{-K\beta}\right). \tag{3.11}$$

Proof. Substitute $t = u^2$ and $\sqrt{\beta}u - \frac{i\alpha}{2\sqrt{\beta}} = v$. Then we get

$$\frac{\alpha\beta^{-\frac{3}{2}}}{2\pi}\exp\left(-\frac{\alpha^2}{4\beta}\right)\int_{-\infty+i\alpha/2\beta}^{\infty+i\alpha/2\beta} e^{-v^2}\,dv$$

and this immediately implies (3.11). □

Applying this lemma to (3.10) yields the main term of (3.7).

3.5. The remainder integrals. In this section we have to show that those parts of the Cauchy integral (3.9), where z or at least one of the u_j lies in Γ_{04} or Γ_{j4}, respectively, are asymptotically negligible. Therefore, let I_p denote the integral (3.10) and R_p the remaining integral. Obviously, we have

$$I_p = \mathcal{O}\left(z_0^{-n}n^{-p-1}\right),\ n\to\infty. \tag{3.12}$$

In order to estimate R_p, observe that for $z\in\Gamma_{04}$ and $u_j\in\Gamma_{j4}$, respectively, the relations

$$|z^{-n-1}| = \mathcal{O}\left(z_0^{-n-1}e^{-\log^2 n}\right) \text{ and } |u_j^{-m_j-1}| = \mathcal{O}\left(e^{-\log^2 m_j}\right)$$

hold. $B(z,u_1,\dots,u_p)$ is composed of $\phi_1(z,u,v)$ and $\phi_2(z,u,v,w)$. As both functions are analytic inside the integration domain (and thus bounded there) and, moreover, the latter one appears only to the first power, it suffices to study the behavior of $\phi_1(z,u,v)$. Inside the domain $\max(|z-z_0|,|u-1|,|v-1|)\le\varepsilon$, $\varepsilon>0$ sufficiently small, we may use the local representation (3.3) provided that ε is sufficiently small. Let $z = 1+\frac{t}{n}$ and consider the expression

$$A = 1 - \frac{\sigma}{\sqrt{2}}\sqrt{-\frac{t}{n}}$$

for $t \in \Gamma_0$ and $z_0|\frac{t}{n}| \le \varepsilon$. If $t \in \Gamma_{01}$, then

$$-t = e^{i\psi}, \quad \psi \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$$

and immediately we get $|A| \le 1$. Let $t \in \Gamma_{02} \cup \Gamma_{03}$, meaning $t = r \pm i$, where $0 \le r \le \log^2 n$. Then

$$\sqrt{-\frac{t}{n}} = \frac{(1+r^2)^{1/4}}{\sqrt{n}} \exp\left(i\left(\frac{\pi}{2} - \frac{1}{2}\arctan\frac{1}{r}\right)\right),$$

and that implies

$$|A|^2 = \begin{cases} 1 - \dfrac{\sigma}{\sqrt{n}} + \mathcal{O}\left(\dfrac{r}{\sqrt{n}}\right) + \mathcal{O}\left(\dfrac{1}{n}\right) & \text{for small } r, \\ 1 - \dfrac{\sigma}{\sqrt{2rn}} + \mathcal{O}\left(\dfrac{1}{\sqrt{r^5 n}}\right) + \mathcal{O}\left(\dfrac{\log^2 n}{n}\right) & \text{for large } r. \end{cases}$$

It remains to investigate the case $z \in \Gamma_{04}$. In this case we have $\frac{z}{z_0} = ae^{i\psi/n}$, where

$$a = \left|1 + \frac{\log^2 n + i}{n}\right|$$

and $\psi \le \varepsilon n$. An easy calculation shows

$$\sqrt{-\frac{t}{n}} \sim \sqrt{-\frac{\log^2 n}{n} - ia\frac{\psi}{n}}$$

and using this we immediately obtain $|A| \le 1$.

Obviously, the above considerations are also valid if we use

$$\sqrt{-\frac{\varphi_0}{\varphi(\tau)}\left(\frac{s_j}{m_j} + \cdots + \frac{s_p}{m_p}\right) - \frac{t}{n}}$$

or sums of terms of this form instead of $\sqrt{-\frac{t}{n}}$. Thus we have for $\max(|z - z_0|, |u-1|, |v-1|) \le \varepsilon$ the inequality

$$|\phi_1(z,u,v)| \le 1$$

which implies

$$R_p = \mathcal{O}\left(z_0^{-n} e^{-C\log^2 n}\right) \tag{3.13}$$

for a suitable constant C.

Now let (z,u,v) be outside the region where the local expansion of $\phi_1(z,u,v)$ is valid. Set $z = z_0\left(1 + \frac{t}{n}\right)$, $u = 1 + \frac{s}{m}$, and $v = 1 + \frac{r}{l}$, l, m proportional to n and, for example, $\left|\frac{r}{l}\right| > \varepsilon$. Then $\phi_1(z,u,v)$ is analytic for $|u| \le \left|1 + cm^{-1/3}\right|$ and $|z| \le z_0\left|1 + c'n^{-1/3}\right|$. Thus it is bounded and, as the exponents k_i and l_i are bounded by $\sqrt{n}$, we have

$$|B(z,u_1,\ldots,u_p)| = \mathcal{O}\left(e^{\sqrt{n}}\right).$$

Conversely, we may choose the circles $|u| = \left|1 + cm^{-1/3}\right|$ and $|z| = z_0 \left|1 + c'n^{-1/3}\right|$ as integration contours for u and z. Thus we get

$$R_p = \mathcal{O}\left(z_0^{-n} \exp\left(\sqrt{n} - n^{2/3}\right)\right). \tag{3.14}$$

Finally, (3.12)–(3.14) imply that the remainder integrals are exponentially small and therefore negligible which completes the proof of Theorem 3.1.

3.6. Tightness. In order to complete the proof of Theorem 1.1 we have to prove that the process $\hat{X}_n(t)$ is tight. This can be done by employing Theorem 12.3 of [4]: The first condition is trivial, as $P\{\hat{X}_n(0) = 0\} = 1$. Furthermore, it can be shown that for polygonal functions like $\hat{X}_n(t)$ it suffices to establish the second condition of this theorem only for the vertices of the polygon (use the ideas of [9, p. 86]); i.e., we have to prove that

$$P\left\{\left|\hat{X}_n\left(\frac{i}{n}\right) - \hat{X}_n\left(\frac{j}{n}\right)\right| \geq \varepsilon\right\} \leq \frac{K}{\varepsilon^\beta}\left|\frac{i-j}{n}\right|^\alpha,$$

where $K > 0$, $\beta \geq \alpha > 1$ holds for all $n \geq 1, 0 \leq i, j \leq n, \varepsilon > 0$. Therefore, we have to set up the GF corresponding to the bivariate distributions of $\hat{X}_n(t)$:

$$\begin{aligned}
B_{k_1k_2}(z, u_1, u_2) &= \sum_{l=0}^{\min(k_1,k_2)-1} B_{k_1lk_2}(z, u_1, u_2) \\
&= \varphi_0^p z^2 u_1 u_2^2 \phi_2(z, u_1u_2, u_2) \sum_{l=0}^{\min(k_1,k_2)-1} \phi_1(z, u_1u_2, 1)^l \\
&\quad \times \phi_1(z, u_1u_2, u_2)^{k_1-1-l} \phi_1(z, u_2, 1)^{k_2-1-l} \\
&= \varphi_0^p z^2 u_1 u_2^2 \phi_2(z, u_1u_2, u_2) \phi_1(z, u_1u_2, u_2)^{k_1-1} \\
&\quad \times \phi_1(z, u_2, 1)^{k_2-1} \frac{1 - q(z, u_1u_2, u_2)^{\min(k_1,k_2)}}{1 - q(z, u_1u_2, u_2)},
\end{aligned}$$

where

$$q(z, u, v) = \frac{\phi_1(z, u, 1)}{\phi_1(z, u, v)\phi_1(z, v, 1)}.$$

Then

$$P\left\{\left|\hat{X}_n\left(\frac{\lfloor \mu_1 n\rfloor}{n}\right) - \hat{X}_n\left(\frac{\lfloor \mu_2 n\rfloor}{n}\right)\right| \geq \varepsilon\right\} = \frac{1}{a_n}[z^n u_1^{m_1} u_2^{m_2}] \sum_{\substack{k,l\geq 1 \\ |k-l|\geq \lfloor \varepsilon\sqrt{n}\rfloor}} B_{kl}(z, u_1, u_2).$$

Therefore, we have to get estimates for the expression

$$\frac{1}{1-q} \sum_{\substack{k,l\geq 0 \\ |k-l|\geq \lfloor \varepsilon\sqrt{n}\rfloor}} x^k y^l - \frac{1}{1-q} \sum_{\substack{k,l\geq 0 \\ |k-l|\geq \lfloor \varepsilon\sqrt{n}\rfloor}} x^k y^l q^{\min(k,l)+1},$$

where we used the abbreviations

$$x = \phi_1(z, u_1u_2, u_2), \quad y = \phi_1(z, u_2, 1), \quad xyq = \phi_1(z, u_1u_2, 1).$$

Splitting this sum yields

$$S_1 = \sum_{k \ge \lfloor \varepsilon\sqrt{n} \rfloor} \sum_{l < k - \lfloor \varepsilon\sqrt{n} \rfloor} x^k y^l = \frac{x^{1+\lfloor \varepsilon\sqrt{n} \rfloor}}{(1-x)(1-xy)},$$

$$S_2 = \sum_{k \ge 0} \sum_{l \ge k + \lfloor \varepsilon\sqrt{n} \rfloor} x^k y^l = \frac{y^{\lfloor \varepsilon\sqrt{n} \rfloor}}{(1-y)(1-xy)},$$

$$S_3 = q \sum_{k \ge \lfloor \varepsilon\sqrt{n} \rfloor} \sum_{l < k - \lfloor \varepsilon\sqrt{n} \rfloor} x^k (qy)^l = \frac{qx^{1+\lfloor \varepsilon\sqrt{n} \rfloor}}{(1-x)(1-xyq)},$$

$$S_4 = q \sum_{k \ge 0} \sum_{l \ge k + \lfloor \varepsilon\sqrt{n} \rfloor} (xq)^k y^l = \frac{qy^{\lfloor \varepsilon\sqrt{n} \rfloor}}{(1-y)(1-xyq)}.$$

Summing up gives

$$\frac{S_1 + S_2 - S_3 - S_4}{1-q} = \frac{x^{1+\lfloor \varepsilon\sqrt{n} \rfloor}}{(1-x)(1-xy)(1-xyq)} + \frac{y^{\lfloor \varepsilon\sqrt{n} \rfloor}}{(1-y)(1-xy)(1-xyq)}. \tag{3.15}$$

Now we are ready to estimate the coefficient $[z^n u_1^{m_1} u_2^{m_2}]$ of (3.15). If we substitute $u_1 u_2 = u, u_2 = v$ and calculate the coefficient $[z^n u^m v^l]$, $m = \mu n, l = \lambda n$, then v keeps track of the difference $l = (i-j)$ which is the most important quantity in proving tightness after all. As the terms $S_1 - S_3$ and $S_2 - S_4$ are of similar form, it suffices to consider one of those, say, $S_1 - S_3$. In order to get an estimate we again use Cauchy's integral formula:

$$[z^n u^m v^l](S_1 - S_3) = \frac{1}{(2\pi i)^3} \int_{\Gamma_z} \int_{\Gamma_u} \int_{\Gamma_v} \times \frac{\phi_1(z,u,v)^k z^{-n-1} u^{-m-1} v^{-l-1} \, dv \, du \, dz}{(1-\phi_1(z,u,v))(1-\phi_1(z,u,v)\phi_1(z,v,1))(1-\phi_1(z,u,1))}, \tag{3.16}$$

where $k = \lfloor \varepsilon\sqrt{n} \rfloor$ and the integration contour $\Gamma_z = \Gamma_{z1} \cup \Gamma_{z2} \cup \Gamma_{z3} \cup \Gamma_{z4}$ is chosen as shown in Figure 3.5. If z is sufficiently close to z_0, meaning that the local representation (3.2) holds, then the integration in u and v is done along the analogous contours (i.e., z_0 has to be replaced with $\tilde{f}(z)$, for Γ_u we replace $\frac{t}{n}$ with $\frac{s}{m}$, and for Γ_v we use $\frac{r}{l}$ instead of $\frac{t}{n}$, where $m = \mu n$ and $l = \lambda n$). Otherwise we choose the unit circle as the integration contour for u und v. To proceed we need the following result.

Lemma 3.5. *Let $f_n \ge 0$ and*

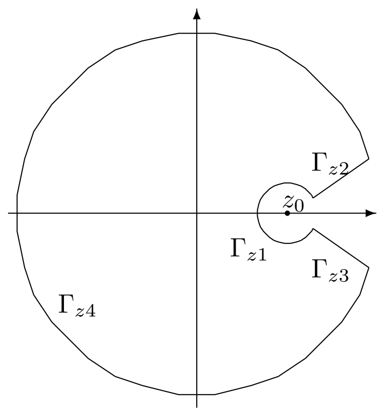
$$F(z) = \sum_{n \ge 0} f_n z^n.$$

Assume that $F(z)$ is analytic in the domain

$$\Delta = \{ z \mid |z| \le 1 + \varepsilon, |\arg(z-1)| \ge \alpha \},$$

$\varepsilon > 0$, and satisfies for $z \in \Delta$ the inequality

$$|F(z)| \le \left| e^{-C\sqrt{1-z}} \right|,$$



$$\Gamma_{z1} = \left\{ z = z_0 \left(1 + \frac{e^{it}}{n}\right) \middle| \alpha \le |t| \le \pi \right\},$$

$$\Gamma_{z2} = \left\{ z = z_0 \left(1 + \frac{t}{n} e^{i\alpha}\right) \middle| 1 \le |t| \le \log^2 n \right\},$$

$$\Gamma_{z3} = \overline{\Gamma}_{z2}$$

$$\Gamma_{z4} = \left\{ z \,\middle|\, |z| = z_0 \left| 1 + \frac{\log^2 n}{n} e^{i\alpha} \right|, \right.$$

$$\left. \arg\left(1 + \frac{\log^2 n}{n} e^{i\alpha}\right) \le |\arg z| \le \pi \right\}.$$

FIG. 3.5. *Integration contour Γ_z.*

where $C > 0$. Then there exists a constant $C' > 0$ such that

$$[z^n]F(z)^k = \mathcal{O}\left(\frac{1}{n} \exp\left(-C' \frac{k}{\sqrt{n}}\right)\right) \tag{3.17}$$

uniformly for $k \ge 0$.

Proof. For convenience assume $z_0 = 1$. Furthermore, C_i will denote appropriate positive constants throughout this proof. We have

$$[z^n]F(z)^k = \frac{1}{2\pi i} \int_{\Gamma_z} \frac{F(z)^k}{z^{n+1}} \, dz.$$

First let $z \in \Gamma_{z1}$. Obviously, the relations

$$\Re\sqrt{1 - z} \ge \frac{C_1}{\sqrt{n}} \tag{3.18}$$

and $z^{-n-1} = \mathcal{O}(1)$ hold. The length of the integration contour is $\mathcal{O}\left(\frac{1}{n}\right)$ and thus

$$\int_{\Gamma_{z1}} \frac{F(z)^k}{z^{n+1}} \, dz = \mathcal{O}\left(\frac{1}{n} \exp\left(-C_2 \frac{k}{\sqrt{n}}\right)\right).$$

Now let $z \in \Gamma_{z2} \cup \Gamma_{z3}$. Then $z = 1 + \frac{t}{n} e^{i\alpha}$ and thus

$$z^{-n-1} = \mathcal{O}\left(\exp\left(-te^{i\alpha}\right)\right)$$

holds. The estimate (3.18) is also valid. To get the desired result we extend the integration contour to $0 \le t \le \infty$. This leads to

$$\int_{\Gamma_{z2} \cup \Gamma_{z3}} \frac{F(z)^k}{z^{n+1}} \, dz \le \frac{C_3}{n} \int_0^\infty \exp\left(-C_4 \frac{k}{\sqrt{n}} - C_5 t\right) \, dt$$

$$= \frac{C_3}{nC_5} \exp\left(-C_4 \frac{k}{\sqrt{n}}\right).$$

Finally, let $z \in \Gamma_{z4}$. Obviously, (3.18) still holds and thus

$$|F(z)|^k \leq \exp\left(-C_6 \frac{k}{\sqrt{n}}\right).$$

Additionally, we have

$$|z|^{-n-1} \sim e^{-\log^2 n} \leq \frac{1}{n}$$

and this implies

$$\int\limits_{\Gamma_{z4}} \frac{F(z)^k}{z^{n+1}}\, dz = \mathcal{O}\left(\frac{1}{n} \exp\left(-C_6 \frac{k}{\sqrt{n}}\right)\right).$$

Finally, set $C' \leq \min(C_2, C_4, C_6)$ to get (3.17). □

Now we are able to estimate the integral (3.17). If (z,u,v) lies in the domain where the local expansion (3.2) holds, then we may estimate the denominator of the integrand as follows:

$$\begin{aligned}
&|(1-\phi_1(z,u,v))(1-\phi_1(z,u,v)\phi_1(z,v,1))(1-\phi_1(z,u,1))| \\
&\quad = \bar{C}^3 \left(\sqrt{1-\frac{u}{\tilde{f}(z)}} + \sqrt{1-\frac{v}{\tilde{f}(z)}}\right) \left(\sqrt{1-\frac{u}{\tilde{f}(z)}} + \sqrt{1-\frac{1}{\tilde{f}(z)}}\right) \\
&\qquad \times \left(\sqrt{1-\frac{u}{\tilde{f}(z)}} + 2\sqrt{1-\frac{v}{\tilde{f}(z)}} + \sqrt{1-\frac{1}{\tilde{f}(z)}}\right) \\
&\quad \geq \frac{C_1}{n^{3/2}} \left(\frac{1}{\sqrt{\mu}} + \frac{1}{\sqrt{\lambda}}\right) \left(1 + \frac{1}{\sqrt{\mu}} + \frac{2}{\sqrt{\lambda}}\right) \left(1 + \frac{1}{\sqrt{\mu}}\right) \geq \frac{C_2}{n^{3/2}},
\end{aligned}$$

where $\bar{C} = \frac{\sigma}{\sqrt{2}} \sqrt{z_0 \frac{\varphi_0}{\varphi(\tau)}}$ and λ, μ are as defined above. It is an easy exercise to verify the validity of the above inequality for the whole integration domain. In order to cope with the numerator we have to distinguish two cases: $\hat{X}_n(t)$ for t close to $t = \frac{\varphi_0}{\varphi(\tau)}$ and for t away from $t = \frac{\varphi_0}{\varphi(\tau)}$.

3.6.1. The process $\hat{X}_n(t)$ outside the vicinity of $t = \frac{\varphi_0}{\varphi(\tau)}$. Let us further consider the domain where (3.2) holds and substitute $\bar{u} = \frac{u}{\tilde{f}(z)}$, $\bar{v} = \frac{v}{\tilde{f}(z)}$ in (3.17). From (3.2) we obviously get

$$\phi_1(z,u,v) = \mathcal{O}\left(\left|\exp\left(-C\left(\sqrt{1-\bar{u}} + \sqrt{1-\bar{v}}\right)\right)\right|\right) \tag{3.19}$$

and application of Lemma 3.5 yields the upper bound

$$\frac{C_1}{ml} \exp\left(-C_2\left(\frac{k}{\sqrt{m}} + \frac{k}{\sqrt{l}}\right)\right) \int \frac{|\tilde{f}(z)|^{-l-m}}{|z^{n+1}|}\, dz \tag{3.20}$$

for the integral (3.17).

In order to proceed we expand $\tilde{f}$ in a Taylor series and get

$$\begin{aligned}
\frac{1}{\tilde{f}(z)} &= \frac{1}{\tilde{f}(z_0)} - \frac{\tilde{f}'(z_0)}{\tilde{f}(z_0)^2}(z-z_0) + \mathcal{O}\left((z-z_0)^2\right) \\
&= 1 + \frac{\varphi(\tau)}{\varphi_0}\left(\frac{z}{z_0} - 1\right) + \mathcal{O}\left((z-z_0)^2\right).
\end{aligned} \tag{3.21}$$

Using $\frac{z}{z_0} = 1 + \frac{e^{it}}{n}$ for $z \in \Gamma_{z1}$ and $\frac{z}{z_0} = 1 + \frac{t}{n}e^{i\alpha}$ for $z \in \Gamma_{z2} \cup \Gamma_{z3}$ we obtain

$$\tilde{f}(z)^{-m-l}\left(\frac{z}{z_0}\right)^{-n} \sim \begin{cases} \exp\left(\left(\frac{\varphi(\tau)}{\varphi_0}(\lambda+\mu)-1\right)e^{it}\right) = \mathcal{O}(1) & \text{for } z \in \Gamma_{z1}, \\ \exp\left(\left(\frac{\varphi(\tau)}{\varphi_0}(\lambda+\mu)-1\right)te^{i\alpha}\right) & \text{for } z \in \Gamma_{z2} \cup \Gamma_{z3}. \end{cases} \tag{3.22}$$

Under the assumption $\frac{\varphi(\tau)}{\varphi_0}(\lambda+\mu) \le 1-\eta$, $\eta > 0$, this implies

$$\begin{aligned} \int\limits_{\Gamma_{z1}\cup\Gamma_{z2}\cup\Gamma_{z3}} \frac{|\tilde{f}(z)|^{-l-m}}{|z^{n+1}|}\,|dz| &= \mathcal{O}\left(\frac{1}{n}z_0^n\left(1+\int\limits_0^\infty \exp\left(-\eta t\cos\alpha\right)\,dt\right)\right) \\ &= \mathcal{O}\left(\frac{1}{n}z_0^n\right). \end{aligned} \tag{3.23}$$

It remains to consider $z \in \Gamma_{z4}$. As long as (z,u,v) lies inside a sufficiently small δ-ball U_δ around the singularity we may still use (3.2). Set $\frac{z}{z_0} = ae^{it/n}$, where $a = \left|1 + \frac{\log^2 n}{n}e^{i\alpha}\right|$ and $|t| \le \delta n$. Then we have

$$|\tilde{f}(z)| \ge 1 - \frac{\varphi(\tau)}{\varphi_0}\frac{\log^2 n}{n} + \frac{t}{n}$$

and therefore $\frac{\varphi(\tau)}{\varphi_0}(\lambda+\mu) \le 1-\eta$ yields

$$\begin{aligned} |\tilde{f}(z)|^{-m-l}\left(\frac{z}{z_0}\right)^{-n} &\le |\tilde{f}(z)|^{-\frac{\varphi_0}{\varphi(\tau)}n+\eta n}\left(\frac{z}{z_0}\right)^{-n} \\ &\le \exp\left(-\eta\frac{\varphi(\tau)}{\varphi_0}\log^2 n - \left(\frac{\varphi_0}{\varphi(\tau)}-\eta\right)t\right) \end{aligned} \tag{3.24}$$

and therefore

$$\int\limits_{\Gamma_{z4}\cap U_\delta} \frac{|\tilde{f}(z)|^{-l-m}}{|z^{n+1}|}\,|dz| = \mathcal{O}\left(e^{-\log^2 n}\right). \tag{3.25}$$

If $(z,u,v) \notin U_\delta$, then the inequality

$$|\phi_1(z,u,v)| \le 1-\vartheta$$

with $\vartheta > 0$ holds. Thus the corresponding integral is exponentially small and therefore negligible.

Collecting (3.20), (3.23), (3.25), and recalling $k = \lfloor \varepsilon\sqrt{n}\rfloor$ we obtain for $\lambda+\mu \le \frac{\varphi_0}{\varphi(\tau)} - \eta$

$$\begin{aligned} &\left|[z^n u^m v^l](S_1 - S_3)\right| \\ &\quad = \frac{1}{(2\pi)^3}\int\limits_{\Gamma_z}\int\limits_{\Gamma_u}\int\limits_{\Gamma_v}\left|\frac{\phi_1(z,u,v)^k z^{-n-1}u^{-m-1}v^{-l-1}\,dv\,du\,dz}{(1-\phi_1(z,u,v))(1-\phi_1(z,u,v)\phi_1(z,v,1))(1-\phi_1(z,u,1))}\right| \\ &\quad \le C\frac{z_0^n n^{3/2}}{nml}\exp\left(-D\left(\frac{k}{\sqrt{m}}+\frac{k}{\sqrt{l}}\right)\right) \\ &\quad \le C\frac{z_0^n n^{-3/2}}{\varepsilon^4}\left(\frac{\varepsilon}{\sqrt{\mu}}\frac{\varepsilon}{\sqrt{\lambda}}\right)^2\exp\left(-D\left(\frac{\varepsilon}{\sqrt{\mu}}+\frac{\varepsilon}{\sqrt{\lambda}}\right)\right), \end{aligned} \tag{3.26}$$

where C and D are suitably chosen positive constants. Applying (3.1) and $x^2 e^{-x} \le \frac{c}{x^k}$, for arbitrary $k > 0$ and sufficiently large $c > 0$, we obtain the tightness condition

$$\begin{aligned} P\{|\hat{X}_n(s) - \hat{X}_n(t)| \ge \varepsilon\} &\le C \frac{1}{\varepsilon^6} \left(\frac{\varepsilon}{\sqrt{|s-t|}} \right)^2 \exp\left(-D \frac{\varepsilon}{\sqrt{|s-t|}} \right) \\ &\le \frac{C'}{\varepsilon^{12}} |s-t|^{3/2}, \end{aligned} \tag{3.27}$$

for $0 < s, t < \frac{\varphi_0}{\varphi(\tau)} - \eta$.

3.6.2. The process $\hat{X}_n(t)$ in the vicinity of $t = \frac{\varphi_0}{\varphi(\tau)}$. To prove the tightness inequality in the case of $\lambda + \mu \ge \frac{\varphi_0}{\varphi(\tau)} - \eta$ it suffices to investigate the terms (3.22) and (3.24). Integrating the right-hand sides of these formulae with respect to t gives, as in the derivation of (3.26), the estimate

$$\left| [z^n u^m v^l](S_1 - S_3) \right| \le C \frac{z_0^n n^{3/2}}{\left(n - \frac{\varphi(\tau)}{\varphi_0}(m+l) \right) ml} \exp\left(-D \left(\frac{k}{\sqrt{m}} + \frac{k}{\sqrt{l}} \right) \right)$$

which directly yields the tightness condition (3.27) if $n - \frac{\varphi(\tau)}{\varphi_0}(m+l) \ge \frac{\varphi(\tau)}{\varphi_0} l$ holds.

So let $n - \frac{\varphi(\tau)}{\varphi_0}(m+l) \le \frac{\varphi(\tau)}{\varphi_0} l$. If we prove

$$P\left\{ \hat{X}_n \left(\frac{\varphi_0}{\varphi(\tau)} - \delta \right) \ge \varepsilon \right\} \le \frac{C}{\varepsilon^\gamma} \delta^\alpha \tag{3.28}$$

with $\gamma \ge \alpha > 1$, then by means of

$$P\left\{ |\hat{X}_n(\mu) - \hat{X}_n(\mu_2)| \ge \varepsilon \right\} \le P\left\{ \hat{X}_n(\mu) \ge \frac{\varepsilon}{2} \right\} + P\left\{ \hat{X}_n(\mu_2) \ge \frac{\varepsilon}{2} \right\}$$

and

$$\begin{aligned} \frac{\varphi_0}{\varphi(\tau)} - \mu - \lambda \le \lambda \quad \text{and} \\ \frac{\varphi_0}{\varphi(\tau)} - \mu_2 \le \lambda, \end{aligned}$$

the tightness condition can be immediately established. To prove (3.28), again set $k = \lfloor \varepsilon \sqrt{n} \rfloor$. Then we have

$$P\left\{ \hat{X}_n \left(\frac{m}{n} \right) \ge \varepsilon \right\} = \frac{\varphi_0}{a_n} [z^{n-1} u^{m-1}] \frac{\phi_1(z,u,1)^k}{1 - \phi_1(z,u,1)}.$$

Using the same integration contour as in the previous section and the substitution $\frac{u}{f(z)} = \bar{u}$ we obtain, as before,

$$|1 - \phi_1(z,u,1)| \ge \frac{C}{\sqrt{n}}$$

for a suitable positive constant C. Furthermore, we have

$$\left(\frac{z}{z_0} \right)^n \sim \left(\frac{1}{f(z)} \right)^{n\varphi_0/\varphi(\tau)}$$

and of course (3.19). Thus, substituting $x = \frac{1}{f(z)}$ gives, finally,

$$[z^{n-1}u^{m-1}]\frac{\phi_1(z,u,1)^k}{1-\phi_1(z,u,1)} \le C_1 \int\int \left|e^{-k\sqrt{1-\bar{u}}}\bar{u}^{-m}\right| \left|e^{-k\sqrt{1-x}}x^{-\frac{\varphi_0}{\varphi(\tau)}n+m}\right| |d\bar{u}\,dx|$$
$$\le \frac{C_2 k^4 n^{-3/2}}{\varepsilon^4\left(\frac{\varphi_0}{\varphi(\tau)}n-m\right)m}\exp\left(-C_3\left(\frac{k}{\sqrt{m}}+\frac{k}{\sqrt{\frac{\varphi_0}{\varphi(\tau)}n-m}}\right)\right),$$

where C_1, C_2, C_3 are appropriate constants. This implies tightness and thus the proof of Theorem 1.1 is complete.

Now we are able to complete the proof of Theorem 3.2. We have only to show that the difference of the contour process and the step process $\hat{X}_n(\lfloor t\rfloor/n)$ converges to zero in probability. Obviously, we have for $t \in [i/n, (i+1)/n]$

$$\left|\hat{X}_n(t) - \hat{X}_n\left(\frac{i}{n}\right)\right| \le \left|\hat{X}_n\left(\frac{i+1}{n}\right) - \hat{X}_n\left(\frac{i}{n}\right)\right|.$$

Combining this with the tightness inequality (3.27) we get

$$P\left\{\left|\hat{X}_n(t) - \hat{X}_n\left(\frac{i}{n}\right)\right| \ge \varepsilon\right\} \le \frac{C'}{\varepsilon^{12}}n^{-3/2}$$

which proves the theorem.

4. The traverse process. In order to deal with the traverse process we first have to set up the basic GFs. The procedure is analogous to that used in the previous section: we mark the nodes associated with the vertices of the polygonal functions of which the process is constructed. Then the nodes of all subtrees to the left of that one which contains a marked node contribute the term 2 to the number of the considered node as each edge is passed twice during pre-order traversal. Thus the GF is given by

$$\tilde{y}(z,u) = a(zu^2) = y(zu^2, 1).$$

From Lemma 3.1 we immediately get the local expansion

$$a(zu^2) \sim \tau - \sqrt{\frac{2\varphi(\tau)}{\varphi''(\tau)}}\sqrt{-\frac{t}{n}-\frac{2s}{m}}$$

for $z = z_0\left(1+\frac{t}{n}\right)$, $u = 1+\frac{s}{m}$, and $n, m \to \infty$ where $m \sim cn, c > 0$.

Remark. Note that we did not define the traverse process on the tree T but instead on $T' = \{\circ\} \times T$ in order to avoid zeros away from the boundary. This modification only causes a factor z in the GFs and a layer shift.

Let a_{mn} denote the sum of weights over all trees where the mth node of the traverse function coincides with the root and let

$$A(z,u) = \sum_{m,n\ge 0} a_{mn}z^n u^m$$

be the associated GF. Suppose that the root has degree i. Obviously, the path of the traverse function passes the root if and only if j $(0 \le j \le i)$ trees have already been traversed completely, but no node of the $(j+1)$st tree has been visited. This implies

$$\begin{aligned}\tilde{A}(z,u) &= uz\sum_{i\geq 0}\varphi_i\sum_{j=0}^{i}\tilde{y}(z,u)^j\tilde{y}(z,u)^{i-j}\\ &= uz\frac{\tilde{y}(z,u)\varphi(\tilde{y}(z,u))-\tilde{y}(z,1)\varphi(\tilde{y}(z,1))}{\tilde{y}(z,u)-\tilde{y}(z,1)}.\end{aligned}$$

Define

$$\tilde{\phi}_1(z,u,v) = uvz\frac{\varphi(\tilde{y}(z,u))-\varphi(\tilde{y}(z,v))}{\tilde{y}(z,u)-\tilde{y}(z,v)}$$

and $\tilde{\phi}_2$ analogously to ϕ_2. It is also easy to see from the previous section that an analogue of Lemma 3.3 applies and thus we are able to set up the GF leading to the finite-dimensional distributions of the process:

$$\begin{aligned}B(z,u_1,\ldots,u_p) = \prod_{i=1}^{p-1} & A\left(z(u_i\cdots u_p)^2,u_p\right)\left[\phi_1(z,u_i\cdots u_p,u_{i+1}\cdots u_p)^{k_i-l_i-1}\right.\\ &\left.\times\,\phi_1(z,u_i\cdots u_p,1)^{l_i-l_{i-1}-1}\phi_2(z,u_i\cdots u_p,u_{i+1}\cdots u_p)\right]\\ &\times\,A(z,u_p)\phi_1(z,u_p,1)^{k_p-l_{p-1}-1}.\end{aligned}$$

Due to the similarity of the GFs to those associated with the contour process, tightness can be proved analogously which proves Theorem 1.2.

Acknowledgment. The author thanks Michael Drmota for several useful comments and suggestions.

REFERENCES

[1] D. J. Aldous, *The continuum random tree* I, Ann. Probab., 19 (1991), pp. 1–28.
[2] D. J. Aldous, *The continuum random tree* II: *An overview*, in Stochastic Analysis, M. T. Barlow and N. H. Bingham, eds., Cambridge University Press, Cambridge, UK, 1991, pp. 23–70.
[3] D. J. Aldous, *The continuum random tree* III, Ann. Probab., 21 (1993), pp. 248–289.
[4] P. Billingsley, *Convergence of Probability Measures*, John Wiley, New York, 1968.
[5] M. Drmota, *Asymptotic distributions and a multivariate Darboux method in enumeration problems*, J. Combin. Theory Ser. A, 67 (1994), pp. 169–184.
[6] M. Drmota, *The height distribution of leaves in rooted trees*, Discrete Math. Appl., 4 (1994), pp. 45–58 (translated from Diskret. Mat., 6 (1994), pp. 67–82).
[7] M. Drmota and M. Soria, *Marking in combinatorial constructions: Generating functions and limiting distributions*, Theoret. Comput. Sci., 144 (1995), pp. 67–99.
[8] P. Flajolet and A. Odlyzko, *Singularity analysis of generating functions*, SIAM J. Discrete Math., 3 (1990), pp. 216–240.
[9] W. Gutjahr and G. Ch. Pflug, *The asymptotic contour process of a binary tree is a Brownian excursion*, Stochastic Process. Appl., 41 (1992), pp. 69–89.
[10] K. Itô and H. P. McKean, Jr., *Diffusion Processes and Their Sample Paths,* Springer-Verlag, New York, 1965.
[11] A. Meir and J. W. Moon, *On the altitude of nodes in random trees*, Canad. J. Math., 30 (1978), pp. 997–1015.
[12] J. S. Vitter and P. Flajolet, *Average-case analysis of algorithms and data structures*, in Handbook of Theoretical Computer Science, Algorithms and Complexity, Vol. A, J. van Leeuwen, ed., North–Holland, Amsterdam, 1990, pp. 431–524.

SIAM J. DISCRETE MATH.
Vol. 12, No. 4, pp. 459–473

THE COMPACTNESS OF INTERVAL ROUTING*

CYRIL GAVOILLE† AND DAVID PELEG‡

Abstract. The *compactness* of a graph measures the space complexity of its shortest path routing tables. Each outgoing edge of a node x is assigned a (pairwise disjoint) set of addresses, such that the unique outgoing edge containing the address of a node y is the first edge of a shortest path from x to y. The complexity measure used in the context of interval routing is the minimum number of intervals of consecutive addresses needed to represent each such set, minimized over all possible choices of addresses and all choices of shortest paths. This paper establishes asymptotically tight bounds of $n/4$ on the compactness of an n-node graph. More specifically, it is shown that every n-node graph has compactness at most $n/4 + o(n)$, and conversely, there exists an n-node graph whose compactness is $n/4 - o(n)$. Both bounds improve upon known results. (A preliminary version of the lower bound has been partially published in *Proceedings of the* 22*nd International Symposium on Mathematical Foundations of Computer Science*, Lecture Notes in Comput. Sci. 1300, pp. 259–268, 1997.)

Key words. random graphs, shortest path, compact routing tables, interval routing

AMS subject classifications. 05C85, 68Q10, 68R10, 68Q25

PII. S0895480197328631

1. Introduction. An *interval routing scheme* is a way of implementing routing schemes on arbitrary networks. It is based on representing the routing table stored at each node in a compact manner by grouping the set of destination addresses that use the same output port into intervals of consecutive addresses. A possible way of representing such a scheme is to use a connected undirected labeled graph providing the underlying topology of the network. The addresses are assigned to the nodes, and the sets of destination addresses are assigned to each endpoint of the edges. As originally introduced in [17], the scheme required each set of destinations to consist of a single interval. This scheme was subsequently generalized in [18] to allow more than one interval per edge.

Formally, consider an undirected n-node graph $G = (V, E)$. Since G is undirected, each edge $\{u, v\} \in E$ between u and v can be viewed as two arcs, i.e., two ordered pairs, (u, v) and (v, u). The graph G is said to support an *interval routing scheme* (IRS) if there exists a labeling $\mathcal{L}$ of V, which labels every node by a unique integer taken from $\{1, \dots, n\}$, and a labeling $\mathcal{I}$ of the outgoing edges, which labels every exit endpoint of each arc of E by a subset of $\{1, \dots, n\}$, such that between any pair of nodes $x \neq y$ there exists a path $x = u_0, u_1, \dots, u_t = y$ satisfying that $\mathcal{L}(y) \in \mathcal{I}(u_i, u_{i+1})$ for every $i \in \{0, \dots, t-1\}$. The resulting routing scheme, denoted $\mathcal{R} = (\mathcal{L}, \mathcal{I})$, is called a *k-interval routing scheme* (k-IRS) if for every arc (u, v), the collection of labels $\mathcal{I}(u, v)$ assigned to it is composed of at most k intervals of consecutive integers (1 and n being considered as consecutive).

*Received by the editors October 10, 1997; accepted for publication (in revised form) March 22, 1999; published electronically October 19, 1999.

http://www.siam.org/journals/sidma/12-4/32863.html.

†LaBRI, Université Bordeaux I, 351, cours de la Libération, 33405 Talence Cedex, France (gavoille@labri.u-bordeaux.fr).

‡Department of Applied Mathematics and Computer Science, The Weizmann Institute of Science, Rehovot, 76100 Israel (peleg@wisdom.weizmann.ac.il). The research of this author was supported in part by a grant from the Israel Science Foundation and by a grant from the Israel Ministry of Science and Art.

The standard definition of k-IRS assumes a *single* routing path between any two nodes. It therefore forces any two incident arcs $e \neq e'$ to have disjoint labels, i.e., $\mathcal{I}(e) \cap \mathcal{I}(e') = \emptyset$. Here we assume that a given destination may belong to many labels of different arcs incident to a same node. This freedom allows us to implement some adaptive routing schemes and code, for example, the full shortest path information, as does the boolean routing scheme [4]. Our upper and lower bounds apply also to the recent extension of interval routing known as *multidimensional interval routing* [3].

To measure the space efficiency of a given IRS, we use the compactness measure, defined as follows. The *compactness* of a graph G, denoted by IRS(G), is the smallest integer k such that G supports a k-IRS of single shortest paths, that is, a k-IRS that provides only one shortest path between any pair of nodes.

If the degree of every node in G is bounded by d, then a k-IRS for G is required to store at most $O(dk \log n)$ bits of information per node (as each set $\mathcal{I}(e)$ can be coded using $2k \log n$ bits[1]) and $O(km \log n)$ bits in total, where m is the total number of edges of the graph. The compactness of a graph is an important parameter for the general study of the compact routing, whose goal is to design distributed routing algorithms with space-efficient data structures for each router.

Figure 1.1 shows an example of a 2-IRS on a graph G. For instance, arc $(7, 1)$ is assigned two intervals: $\mathcal{I}(7, 1) = \{1, 2, 5\}$. It is easy to verify that this labeling provides a single shortest path between any pair of nodes in G; however, it is more difficult to check whether G has compactness 1. Actually, in [9] it is shown that IRS(G) = 2. Recently, it was proven in [1] that for general graphs, the problem of deciding whether IRS(G) = 1 is NP-complete.

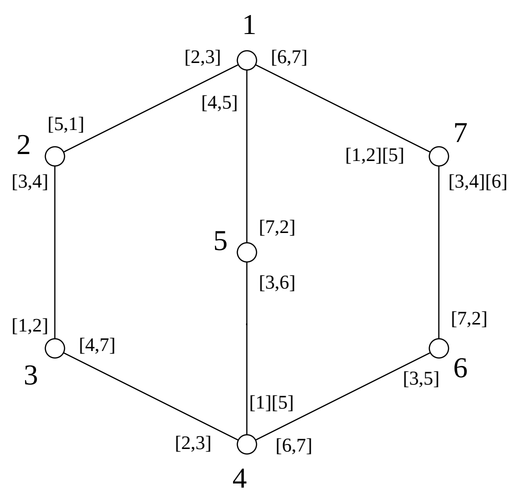


FIG. 1.1. *A 2-IRS for a graph G.*

The compactness of many graph classes has been studied. Its value is 1 for trees [17], outerplanar graphs [6], hypercubes and meshes [18], r-partite graphs [12], interval graphs [16], and unit-circular graphs [5]. It is 2 for tori [18], at most 3 for 2-trees [16], and at most $2\sqrt{n}$ for chordal rings on n nodes [15] (see [7] for a survey of the recent state of the art). Finally, it has been proved that compactness $\Theta(n)$ might be required [9].

[1]A more accurate coding allows us to use only $O(dk \log (n/k))$ bits per node; cf. [7].

The next section presents the results of the paper. In section 3 we prove that $n/4 + o(n)$ intervals are always sufficient, and in section 4 that $n/4 - o(n)$ intervals might be required. We conclude in section 5.

2. The results. Clearly, the compactness of a graph cannot exceed $n/2$, since any set $\mathcal{I}(e) \subset \{1, \ldots, n\}$ containing more than $n/2$ integers must contain at least two consecutive integers, which can be merged into the same interval. On the other hand it has been proved in [9] that for every $n \geq 1$ there exists an n-node graph of compactness at least $n/12$ and $n/8$ for every n power of 2.

In this paper we close this gap by showing that $n/4$ is asymptotically a tight bound for the compactness of n-node graphs. More specifically, we have the following theorem.

THEOREM 2.1. *Every n-node graph G satisfies*

$$\mathrm{IRS}(G) < \frac{n}{4} + \frac{1}{4}\sqrt{2n \ln (3n^2)}.$$

THEOREM 2.2. *For every sufficiently large integer n, there exists an n-node graph G such that*

$$\mathrm{IRS}(G) > \frac{n}{4} - 1.72\, n^{2/3} \ln^{1/3} n.$$

Moreover, G has diameter 2, maximum degree at most $n/2$, and fewer than $1.15\, n^{5/3} \ln^{1/3} n$ edges, and every single k-IRS on G with $k < \mathrm{IRS}(G)$ contains some routing path of length at least 3.

We later show that both the upper and the lower bounds hold even if the single and/or shortest path assumptions are relaxed.

Theorem 2.1 directly improved the results of [5, Theorem 11] and [3, Theorem 2] and also a result of [2, Theorem 9].

The lower bound is proved using Kolmogorov complexity. As a result, only the existence of such a worst-case graph G can be proved. Moreover, the bound gives an asymptotic bound since the Kolmogorov complexity is defined up to a constant. This is in contrast to the technique of [9], which gave explicit recursive constructions of worst-case graphs of compactness $n/12$, for every $n \geq 1$.

3. The upper bound. The basic idea for the upper bound, and partially for the lower bound, is to give a boolean matrix representation $M(\mathcal{R})$ for a given k-IRS $\mathcal{R} = (\mathcal{L}, \mathcal{I})$ on a graph $G = (V, E)$. Recall that for each arc e, $\mathcal{I}(e)$ is the set of addresses that labels the arc e. Let u_e be the characteristic sequence of the subset $\mathcal{I}(e)$ in $\{1, \ldots, n\}$, namely, the ith element of u_e is 1 if $i \in \mathcal{I}(e)$ and 0 otherwise. It is easy to see that there is a one-to-one correspondence between the intervals of $\mathcal{I}(e)$ and the blocks of consecutive ones in u_e. The number of blocks of consecutive ones in u_e can be seen as the occurrence number of 01-sequences[2] in the binary vector u_e. By collecting all the u_e's sequences in order to form a boolean matrix $M(\mathcal{R})$ of dimensions $n \times 2|E|$, the problem of finding a node-labeling $\mathcal{L}$ of G such that each set $\mathcal{I}(e)$ is composed of at most k intervals is equivalent to the problem of finding a row permutation of $M(\mathcal{R})$ such that every column has at most k blocks of consecutive ones.

[2] If u_e does not contain any 0, u_e is composed of exactly one block of consecutive ones.

Throughout this section, M denotes a boolean matrix of n rows and p columns. For every column u of M and for every row permutation π, we denote by $c(u,\pi)$ the number of blocks of consecutive ones in the column u under π. For every matrix M, define the *compactness* of M, denoted $\mathrm{comp}(M)$, as the smallest integer k such that there exists a row permutation π of M satisfying, for every column u of M, $c(u,\pi) \le k$.

The following theorem is the key to the proof of Theorem 2.1.

THEOREM 3.1. *Let M be an $n \times p$ boolean matrix, $p < e^{n/2}/n$, let u be a column of M, and let $A_u(k) = \{\pi \mid c(u,\pi) = k\}$ be the set of row permutations of M that provides k blocks of consecutive ones for the column u. Then for every integer k in the range $n/4 + (1/4)\sqrt{2n \ln(pn)} < k \le n/2$,*

$$|A_u(k)| < \frac{4n!}{pn} .$$

Proof. Let us consider a column u of M and an integer k. Let a (resp., b) be the number of 0's (resp., 1's) of u. Clearly, if $a < k$ or $b < k$, the theorem holds because in this case $A_u(k) = \emptyset$. Hence suppose $a, b \ge k$, with $a + b = n$. There are $a!$ permutations of the rows $\{x_1, \dots, x_a\}$ containing 1 and $b!$ permutations of the rows $\{y_1, \dots, y_b\}$ containing 0 in u, and each such pair of permutations creates a different and disjoint set of permutations in $A_u(k)$. Moreover, each of the $a!$ permutations needs to be broken into k nonempty blocks, which can be done in $\binom{a}{k}$ ways, and similarly for the $b!$ permutations of the rows $\{y_1, \dots, y_b\}$. Each partitioned pair merges, alternating a block of 1's and a block of 0's, in order to yield a permutation in $A_u(k)$. Overall, $|A_u(k)| = a!\binom{a}{k} b!\binom{b}{k}$, and we need to show that

$$a!\binom{a}{k} b!\binom{b}{k} \;<\; \frac{4n!}{pn} . \tag{3.1}$$

Using formula (9.91) of [11, p. 481], derived from Stirling's formula, we have for every $n \ge 1$,

$$\left(\frac{n}{e}\right)^n \sqrt{2\pi n} \;<\; n! \;<\; \gamma \left(\frac{n}{e}\right)^n \sqrt{2\pi n} ,$$

where $\gamma = e^{1/12 - 1/360 + 1/1260} \approx 1.08$. Thus

$$\frac{4n!}{pn} \;>\; \frac{n^n e^{-n} 4\sqrt{2\pi n}}{pn} \;=\; e^{-n}\, 4\sqrt{2\pi}\; \frac{n^{n-1/2}}{p} . \tag{3.2}$$

From Stirling's bound, for every k in the range $0 < k < a$,

$$\binom{a}{k} \;<\; \left(\frac{a}{k}\right)^k \left(\frac{a}{a-k}\right)^{a-k} \frac{\gamma}{\sqrt{2\pi}} \sqrt{\frac{a}{k(a-k)}} . \tag{3.3}$$

This bound cannot apply for $k = a$. Let us first handle the extremal cases.

CLAIM 3.1. *Inequality* (3.1) *holds for $a = k$ or $b = k$ for every integer k, $0 \le k \le n/2$.*

Proof. In both cases assumed in the claim, inequality (3.1) is equivalent to

$$k!(n-k)!\binom{n-k}{k} \;=\; \frac{(n-k)!^2}{(n-2k)!} \;<\; \frac{4n!}{pn^{3/2}} . \tag{3.4}$$

The ratio $(n-k)!^2/(n-2k)!$ increases for $0 \le k \le n/2$. Indeed, in this range $n-k \ge n-2k$; hence $(n-k)! \ge (n-2k)!$, and therefore $(n-k)!^2 \ge (n-2k)!$. It is thus sufficient to prove inequality (3.4) for $k = n/2$, in which case it becomes

$$\left(\frac{n}{2}\right)!^2 < \frac{4n!}{pn}. \tag{3.5}$$

Using Stirling's bound, $(n/2)!^2 < (n/2)^n e^{-n}\gamma^2 2\pi n$, and simplifying with the lower bound of inequality (3.2), we get that to prove inequality (3.5) it suffices to prove

$$\left(\frac{n}{2}\right)^n < c\,\frac{n^{n-3/2}}{p} \quad \text{or} \quad pn^{3/2} < c\,2^n, \quad \text{where } c = \frac{4}{\gamma^2\sqrt{2\pi}} \approx 1.36.$$

This last inequality is satisfied for every $n \ge 1$, since $p < e^{n/2}/n$, and $e^{n/2}\sqrt{n} < c\,2^n$ is equivalent to $n/2 < n(\ln 2 - (1/(2n))\ln n) + \ln c$, which is trivial because $(1/(2n))\ln n \le (1/4)\ln 2$, and $(1-(1/4))\ln 2 \approx 0.51 > 1/2$ and moreover $\ln c > 0$. This completes the proof of Claim 3.1. ∎

For the remainder of the proof, let us assume that $k < a, b$. Therefore, it is possible to apply the bound of inequality (3.3), which gives

(3.6)
$$a!\binom{a}{k} b!\binom{b}{k} < a^a\left(\frac{a}{k}\right)^k\left(\frac{a}{a-k}\right)^{a-k} b^b\left(\frac{b}{k}\right)^k\left(\frac{b}{b-k}\right)^{b-k}\frac{ab\,e^{-n}\gamma^4}{k\sqrt{(a-k)(b-k)}}.$$

CLAIM 3.2. *For integers k, a, b, and n such that $0 < k < a$ and $a+b=n$,*

$$\frac{ab}{k\sqrt{(a-k)(b-k)}} < \frac{3\sqrt{3}}{4}\sqrt{n}.$$

Proof. Set $b = n-a$, and let $f(a) = (a-k)(n-a-k)$. Claim 3.2 holds if

$$\frac{ab}{k\sqrt{f(a)}} < \frac{3\sqrt{3}}{4}\sqrt{n}.$$

Observing that $ab \le (a+b)^2/4 = n^2/4$, it suffices to prove that

$$\frac{n^2}{4k\sqrt{f(a)}} < \frac{3\sqrt{3}}{4}\sqrt{n}.$$

Let us lower bound the term $k\sqrt{f(a)}$. Noting that $f(a)$ is symmetric around the point $n/2$, let us assume without loss of generality that $a \le n/2$. In this range $f'(a) = n-2a \ge 0$. Therefore, in the desired range $f(a)$ attains its minimum, where a is minimum, and thus $k\sqrt{f(a)} > k\sqrt{f(k)} = k\sqrt{n-2k}$. Let $f_2(k) = k\sqrt{n-2k}$. ${f_2}'(k) = \sqrt{n-2k} - k/\sqrt{n-2k}$, which is of the same sign that $n-3k$. Hence in the range $0 < k < n/2$, $f_2(k)$ first decreases until its minimum at the point $n/3$, then increases between $n/3$ and $n/2$. Thus, $f_2(k) \ge f_2(n/3) = (n/3)^{3/2}$. Therefore

$$\frac{n^2}{4k\sqrt{f(a)}} < \frac{n^2}{4(n/3)^{3/2}} = \frac{3\sqrt{3}}{4}\sqrt{n},$$

which completes the proof of Claim 3.2. ∎

In view of Claim 3.2, inequality (3.6) becomes

$$a!\binom{a}{k}b!\binom{b}{k} \;<\; a^a\left(\frac{a}{k}\right)^k\left(\frac{a}{a-k}\right)^{a-k} b^b\left(\frac{b}{k}\right)^k\left(\frac{b}{b-k}\right)^{b-k} e^{-n}\,\gamma^4\frac{3\sqrt{3}}{4}\sqrt{n}\ .$$

Simplifying and applying the lower bound of inequality (3.2), we obtain that to prove inequality (3.1) it suffices to show

$$a^a\left(\frac{a}{k}\right)^k\left(\frac{a}{a-k}\right)^{a-k} b^b\left(\frac{b}{k}\right)^k\left(\frac{b}{b-k}\right)^{b-k} \;<\; \frac{16\sqrt{2\pi}}{\gamma^4\,3\sqrt{3}}\ \frac{n^{n-1}}{p}\ .$$

Noting that $16\sqrt{2\pi}/(\gamma^4\,3\sqrt{3}) \approx 5.57$, it remains to prove that

$$p^{-1}n^{n-1}k^{2k}(a-k)^{a-k}(b-k)^{b-k} - a^{2a}b^{2b} > 0. \tag{3.7}$$

Assume that $k_0 < k < a \le n/2 \le b$, with $b = n-a$ and with $k_0 = n/4 + (1/4)\sqrt{2n\ln(pn)}$. The case $b \le a$ is dual, and at most doubles the number of permutations (which is taken in account in the removing of the multiplicative constant 5.57 in inequality (3.7)).

Let $f(a) = p^{-1}n^{n-1}k^{2k}(a-k)^{a-k}(n-a-k)^{n-a-k} - a^{2a}(n-a)^{2(n-a)}$.

To establish inequality (3.7) and complete the proof, it remains only to show the following lemma.

LEMMA 3.2. *$f(a) > 0$ in the range $k_0 < k < a \le n/2$.*

Proof. Write $f(a) = \exp(A) - \exp(B)$, where

$$A = -\ln p + (n-1)\ln n + 2k\ln k + (a-k)\ln(a-k) + (n-a-k)\ln(n-a-k),$$
$$B = 2a\ln a + 2(n-a)\ln(n-a);$$

then $f(a) > 0$ if and only if $A - B > 0$. Letting $f_2(a) = A - B$, it remains to prove that $f_2(a) > 0$ in the range $k_0 < k < a \le n/2$. The first derivative of f_2 is

$$f_2{}'(a) \;=\; \ln\left(\frac{a-k}{n-a-k}\right) + 2\ln\left(\frac{n-a}{a}\right).$$

CLAIM 3.3. *$f_2{}'(a) \le 0$ in the range $k_0 < k < a \le n/2$.*

Proof. It suffices to show that in the range specified in the claim,

$$\frac{a-k}{n-a-k}\left(\frac{n-a}{a}\right)^2 \le 1,$$

or

$$f_3(a) \;\equiv\; (a-k)(n-a)^2 - (n-a-k)a^2 \;\le\; 0.$$

This is shown by noting that $f_3(a)$ is increasing in this range; hence its maximum is attained at the point $a = n/2$, where $f_3(n/2) = 0$. To show that $f_3(a)$ is increasing, we need to show that $f_3{}'(a) = 6a^2 - 6an + n^2 + 2nk \ge 0$ in this range. This is shown by noting that $f_3{}'(a)$ is decreasing in this range; hence its minimum is attained at the point $a = n/2$, where $f_3{}'(n/2) = (2k - n/2)n \ge 0$ since $k > k_0 > n/4$. To show that $f_3{}'(a)$ is decreasing, we need to show that $f_3{}''(a) = 12a - 6n \le 0$ in this range, which is trivial since $a \le n/2$. This completes the proof of Claim 3.3. □

It follows from Claim 3.3 that $f_2(a)$ is decreasing in this range, and hence its minimum is attained at $a = n/2$. Hence in this range,

$$f_2(a) \;\geq\; f_2\left(\frac{n}{2}\right) \;=\; p^{-1}n^{n-1}\left(\frac{n}{2}-k\right)^{n-2k} - \left(\frac{n}{2}\right)^{2n}.$$

Consequently, it remains to prove that $f_2(n/2) > 0$ in the desired range. Simplifying, we need to show that $k^{2k}(n-2k)^{n-2k}2^{n+2k} - pn^{n+1} > 0$ in the range $k_0 < k < n/2$. Writing $k = \alpha n$, we need to prove that

$$\begin{aligned} \left(\alpha^{2\alpha}(1-2\alpha)^{1-2\alpha}2^{1+2\alpha}\right)^n n^n > pn^{n+1} \text{ or} \\ \alpha^{2\alpha}(1-2\alpha)^{1-2\alpha}2^{1+2\alpha} > (pn)^{1/n} \text{ or that} \\ 2\alpha\log\alpha + (1-2\alpha)\log(1-2\alpha) + 1 + 2\alpha > \frac{\log(pn)}{n} \end{aligned}$$

in the range $k_0/n < \alpha < 1/2$ (the function log represents logarithm to base 2). Let $g(\alpha) = 2\alpha\log\alpha + (1-2\alpha)\log(1-2\alpha) + 1 + 2\alpha$. It remains to prove the following claim.

CLAIM 3.4. $g(\alpha) > \log(pn)/n$ *in the range* $k_0/n < \alpha < 1/2$.

Proof. Note that

$$\frac{k_0}{n} = \frac{1}{4} + \frac{1}{4}\sqrt{\frac{2\ln(pn)}{n}}.$$

Therefore, if $p < e^{n/2}/n$, then $k_0/n < 1/2$; thus the range for α is not empty. Moreover,

$$\begin{aligned} g'(\alpha) &= 2\log\alpha - 2\log(1-2\alpha) + 2\,, \\ g''(\alpha) &= \frac{2}{\alpha\ln 2} + \frac{4}{(1-2\alpha)\ln 2}\,, \\ g'''(\alpha) &= -\frac{2}{\alpha^2\ln 2} + \frac{8}{(1-2\alpha)^2\ln 2}\,. \end{aligned}$$

In the range $1/4 < \alpha < 1/2$, let us show that $g'''(\alpha) > 0$. This happens if

$$\begin{aligned} \frac{8}{(1-2\alpha)^2\ln 2} > \frac{2}{\alpha^2\ln 2} \text{ or} \\ 4\alpha^2\ln 2 > (1-2\alpha)^2\ln 2 \text{ or} \\ 2\alpha > 1-2\alpha, \end{aligned}$$

which is trivial since $\alpha > 1/4$. Moreover $g(1/4) = g'(1/4) = 0$ and $g''(1/4) = 16/\ln 2$. Thus we have the following bound for $g(\alpha)$:

$$g(\alpha) \;>\; \frac{g''(1/4)}{2!}\left(\alpha - \frac{1}{4}\right)^2 \;=\; \frac{8}{\ln 2}\left(\alpha - \frac{1}{4}\right)^2.$$

Thus, it suffices to take α such that

$$\begin{aligned} \frac{8}{\ln 2}\left(\alpha - \frac{1}{4}\right)^2 > \frac{\log(pn)}{n} \text{ or} \\ \alpha > \frac{1}{4} + \sqrt{\frac{\ln 2}{8}\frac{\log(pn)}{n}} \;=\; \frac{1}{4} + \frac{1}{4}\sqrt{\frac{2\ln(pn)}{n}} \;=\; \frac{k_0}{n} \end{aligned}$$

to complete the proof of Claim 3.4. □

This completes the proof of Lemma 3.2 and subsequently of Theorem 3.1. □

COROLLARY 3.3. *Let M be an $n \times p$ boolean matrix, $p < e^{n/2}/n$. Then*

$$\mathrm{comp}(M) \ < \ \frac{n}{4} + \frac{1}{4}\sqrt{2n \ln (pn)}.$$

Proof. We need to show that there exists a row permutation π of M, such that $c(u,\pi) < n/4 + (1/4)\sqrt{2n \ln (pn)}$ for every column u of M. Let us set $k_0 = n/4 + (1/4)\sqrt{2n \ln (pn)}$. A permutation π is said to be "bad" if there exists a column u of M such that $c(u,\pi) > k_0$. Let B_u be the set of bad permutations for the column u, i.e.,

$$B_u = \bigcup_{k_0 < k \leq n/2} A_u(k).$$

The entire set of bad permutations for M is $B = \bigcup_u B_u$ over all the p columns of M. Theorem 3.1 implies that for every u,

$$|B_u| \ < \ \frac{(n/2 - k_0) 4 n!}{pn} \ < \ \frac{n!}{p}$$

because $n/2 - k_0 < n/4$. It follows that $|B| < n!$. Therefore, there is at least one "good" permutation for the rows of M, i.e., a permutation providing at most $\lfloor k_0 \rfloor$ blocks of consecutive ones for each of the columns. We conclude by remarking that $\lfloor k_0 \rfloor < k_0$, since $\ln (pn)$ cannot be an integer for integer $pn > 1$. □

Proof of Theorem 2.1. Let us consider any node labeling $\mathcal{L}$ of V and any routing function $\mathcal{R} = (\mathcal{L}, \mathcal{I})$ on G, e.g., a single shortest path routing function. Form the $n \times p$ boolean matrix $M(\mathcal{R})$ as explained earlier. By Corollary 3.3 (which is clearly applicable as $p < e^{n/2}/n$), there exists a row permutation π such that $c(u,\pi) < n/4 + (1/4)\sqrt{2n \ln (pn)}$ for every column u of M. Permute the labeling of the nodes of V according to π to obtain a labeling $\mathcal{L}'$ such that the resulting interval routing scheme, $\mathcal{R}' = (\mathcal{L}', \mathcal{I})$, is a q-IRS for

$$q \ < \ \frac{n}{4} + \frac{1}{4}\sqrt{2n \ln (pn)} \ , \tag{3.8}$$

namely, $\mathcal{R}'$ has fewer than q intervals on each arc. Let us show that only $p \leq 3n$ arcs have to be considered.

In the case of a single IRS, each destination is assigned to a unique set $\mathcal{I}(e)$ in each node. For each node of degree three or less, we consider all its outgoing edges. Consider a node x of degree greater than three, and let I, J, K be the three largest cardinality sets assigned to outgoing edges of x. Assume that the nodes are relabeled using the permutation π in such a way that all the sets I, J, K are composed respectively of i, j, and k intervals. We remark that $i + j + k < 3n/4 + o(n)$ by Corollary 3.3. Hence all the other sets share at most $n/4$ intervals and do not need to be considered.

We complete the proof by plugging $p = 3n$ in inequality (3.8). □

Remark 1. The parameter p of inequality (3.8) represents the total number of arcs we are required to consider. For graphs with fewer edges one can choose $p = 2|E|$, which is better than $3n$ only for graphs of average degree at most 3. Note that there exist some 3-regular graphs of compactness $\Theta(n)$ [10].

Here we give another application of Theorem 3.1.

COROLLARY 3.4. *Let M be an $n \times p$ boolean matrix, $p < e^{n/2}/n^2$, and let π be an arbitrary row permutation of M. With probability at least $1 - n^{-1}$, $c(u, \pi) < n/4 + (1/4)\sqrt{2n \ln(pn^2)}$ for every column u of M.*

Proof. Let M be an $n \times p$ boolean matrix with $p < e^{n/2}/n^2$. Build from M a matrix M' composed of all the p columns of M and completed by $(n-1)p$ other columns, each filled up with 0's. M' has dimensions $n \times pn$. Clearly, the set of "bad" permutations for M' and M is the same. The total set of bad permutations for M' is

$$B \;=\; \bigcup_u B_u(k) \;=\; \bigcup_u \left(\bigcup_{k_0 < k \le n/2} A_u(k) \right),$$

where the union is taken over all the pn columns u of M' and $k_0 = n/4 + (1/4)\sqrt{2n \ln(pn^2)}$. Theorem 3.1 implies that $|B| < n!/n$, noting that $pn < e^{n/2}/n$. We conclude that the number of "good" permutations for M' (hence for M), i.e., providing at most $\lfloor k_0 \rfloor$ blocks of consecutive ones for all the columns, is at least $n! - n!/n$, which is a fraction of $1 - n^{-1}$ of all the row permutations of M. The proof is completed by remarking that $\lfloor k_0 \rfloor < k_0$ for every integer $pn^2 > 1$. $\square$

Therefore, to have a labeling with fewer than $n/4 + O(\sqrt{n \log n})$ intervals on all the edges of G, it suffices to fix a node labeling and a routing function on G, then to randomly permute the n labels of nodes by choosing a random permutation π of $\{1, \ldots, n\}$.

Note that the previous algorithm applies not only to single shortest path routing schemes, but also to any routing scheme implementable by using interval routing schemes. Thus for every IRS on every graph we can relabel the nodes in order to have at most $n/4 + O(\sqrt{n \log n})$ intervals per arc. It is still unknown whether there exists a polynomial time deterministic IRS construction algorithm that guarantees at most $n/4 + o(n)$ intervals per edges.

We do not know whether the upper bound is reached for certain graphs. However, it is well known that some small graphs have compactness strictly greater than $n/4$. In [9], it is shown that the example depicted on Figure 1.1, with 7 nodes and 8 edges, has compactness 2, whereas all graphs of order at most 6 have compactness 1. Note also that the compactness of the Petersen graph is 3, whereas its order is 10 and its size 15.

4. The lower bound. The lower bound idea is based on a representation similar to the one used in the upper bound, namely, a boolean matrix M representation of the k-IRS on G. However, this time we need to show that no row permutation of M yields fewer than k blocks of consecutive ones on all the columns. Furthermore, this must be shown for every choice of shortest routing paths. For instance, every $\sqrt{n} \times \sqrt{n}$ grid has compactness 1, using the standard node labeling and single-bend YX-routing paths. Clearly, a different choice of shortest routing paths would increase the number of intervals per edge. That is why we use smaller matrices, say, of dimensions $|W| \times |A|$, by considering only a subset of nodes, W, and a subset of arcs, A, where the shortest paths between the tails of the arcs of A and the nodes of W are all unique.

Our worst-case graph construction is a function of a boolean matrix M, denoted G_M. For every $p \times q$ boolean matrix M, define the graph G_M as follows. For every $i \in \{1, \ldots, p\}$, associate with the ith row of M a vertex v_i. For every $j \in \{1, \ldots, q\}$, associate with the jth column of M a pair of vertices $\{a_j, b_j\}$ connected by an edge. In addition, for every $i \in \{1, \ldots, p\}$ and $j \in \{1, \ldots, q\}$, if $m_{i,j} = 0$ we add to G_M

an edge connecting v_i to a_j, and otherwise we connect v_i to b_j. Note that the graph obtained from G_M by contracting the edges $\{a_j, b_j\}$, $j \in \{1, \ldots, q\}$ is a complete bipartite graph $K_{p,q}$. It is easy to see that the shortest path from any a_j to any v_i is unique and is determined by the entry $m_{i,j}$ of M.

For integers p, q, let $\mathcal{M}$ be the collection of $p \times q$ boolean matrices having $\lfloor p/2 \rfloor$ 1-entries per column. Let $\mathcal{M}_1$ be the subset of matrices of $\mathcal{M}$ such that all the rows are pairwise noncomplementing and let $\mathcal{M}_2$ be the subset of matrices of $\mathcal{M}$ such that for every pair of columns the $2 \times p$ matrix composed of the pair of columns contains the submatrix[3] $\left[\begin{smallmatrix} 0&0&1&1 \\ 0&1&0&1 \end{smallmatrix}\right]$ up to column permutation. We next use a direct consequence of a result proved recently in [8]. In the following, $f(n) \sim g(n)$ means that $f(n) = g(n) + o(g(n))$.

LEMMA 4.1 (see Gavoille and Gengler [8]). *Let p, q be two sufficiently large integers. If $p = o(2^{q/2})$ and $q = o(2^{p/4})$, then $|\mathcal{M}_1 \cap \mathcal{M}_2| \sim |\mathcal{M}|$.*

Throughout the remainder of the paper, we set $\mathcal{M}_0 = \mathcal{M}_1 \cap \mathcal{M}_2$. We will see later that the graphs G_M built from the matrices $M \in \mathcal{M}_0$ have diameter 2 exactly. Furthermore, almost all matrices are in $\mathcal{M}_0$.

We will see that the compactness of M is a lower bound of the compactness of G_M. Here we give a lower bound of the compactness of matrices of $\mathcal{M}_0$.

LEMMA 4.2. *For every sufficiently large integer p, q such that $3 \log p \leq q \leq 2^{p/5}$, there exists a $p \times q$ boolean matrix M of $\mathcal{M}_0$ of compactness*

$$\mathrm{comp}(M) \;\geq\; \frac{p}{4} - \sqrt{\frac{3p^2}{2q} \ln p + O\left(\frac{p^2}{q} + p \log p\right)}.$$

Proof. We use a counting argument which can be formalized using Kolmogorov complexity (see [14] for an introduction). Basically, the Kolmogorov complexity of an individual object X is the length (in bits) of the smallest program, written in a fixed programming language, which prints X and halts. A simple counting argument allows us to argue that no program of length less than K can print a certain X_0 taken from a set of more than 2^K elements.

Let us begin by showing that the claim of the lemma holds for some matrices of $\mathcal{M}$. For every $M \in \mathcal{M}$, we define $\mathrm{cl}(M)$ to be the subset of the matrices of $\mathcal{M}$ obtained by row permutation of M. We claim that there exists a matrix $M_0 \in \mathcal{M}$ such that all the matrices of $\mathrm{cl}(M_0)$ have Kolmogorov complexity at least $C = \log|\mathcal{M}| - \log(p!) - 3\log p$. Indeed, consider a matrix $M_0 \in \mathcal{M}$ for which there exists a matrix $M_0' \in \mathrm{cl}(M_0)$ of Kolmogorov complexity $C' < C$. Then M_0 may be described by an ordered pair (i_0, M_0'), where i_0 is the index of the row permutation of M_0' into M_0. Such an integer can be coded, in a self-delimiting way, by $\log(p!) + 2\log p + O(1)$ bits. ($2\lceil \log p \rceil$ bits are sufficient to describe p; thus the length of any $i_0 \leq p!$ in a self-delimiting way.) Hence the Kolmogorov complexity of M_0 is at most $C' + \log(p!) + 2\log p + O(1) < \log|\mathcal{M}|$. By the counting argument mentioned earlier, it is impossible for all matrices $M_0 \in \mathcal{M}$ to have such low complexity.

The class $\mathcal{M}$ is of size $|\mathcal{M}| = \binom{p}{\lfloor p/2 \rfloor}^q = \Theta(2^p/\sqrt{p})^q$. By Stirling's formula $\log|\mathcal{M}| = pq - O(q \log p)$ and $\log(p!) = p \log p - O(p)$. Hence,

$$C \;=\; \log|\mathcal{M}| - \log(p!) - 3\log p \;=\; pq - p\log p - O(p + q\log p). \tag{4.1}$$

All the matrices of $\mathcal{M}$ have q columns, each one of Kolmogorov complexity bounded by $p + O(1)$. Therefore there exists a matrix M_0 such that every matrix in $\mathrm{cl}(M_0)$ has

[3] A is a submatrix of B if A can be obtained from B by removing some columns and rows in B.

a column of Kolmogorov complexity at least

$$\frac{C}{q} - 2\log p \;=\; p - \frac{p\log p}{q} - O\left(\frac{p}{q} + \log p\right). \tag{4.2}$$

The term $2\log p$ codes the length of the description of such a column in a self-delimiting way. Define a *deficiency function* as an $\mathbb{N} \mapsto \mathbb{N}$ function such that it is possible to retrieve n and $\delta(n)$ from $n - \delta(n)$ by a self-delimiting program of constant size. From [14, Theorem 2.15, p. 131], every binary string of length p bits and of Kolmogorov complexity at least $p - \delta(p)$ contains at least

$$\frac{p}{4} - \sqrt{(\delta(p) + c)p\,\frac{3}{2}\ln 2} \tag{4.3}$$

occurrences of 01-sequences for any deficiency function δ and some constant c depending on the definition of the Kolmogorov complexity. Since each 01-sequence in a binary string necessarily starts a new block of consecutive ones, we get a lower bound on the number of blocks of consecutive ones for such strings.

By choosing for δ the function $\delta(p) = (p/q)\log p + O(p/q + \log p)$ and by inequality (4.3), it follows that M_0 has compactness

$$\operatorname{comp}(M_0) \;\geq\; \frac{p}{4} - \sqrt{\frac{3p^2}{2q}\ln p + O\left(\frac{p^2}{q} + p\log p\right)}. \tag{4.4}$$

Finally, let us show that the result of the lemma, shown for some matrices in $\mathcal{M}$, holds also for the compactness of some matrices of $\mathcal{M}_0$. From Lemma 4.1, because $p = o(2^{q/2}) = o(p^{3/2})$, we get that $|\mathcal{M}_1| \sim |\mathcal{M}|$. Similarly, $q = O(2^{p/5}) = o(2^{p/4})$; thus $|\mathcal{M}_2| \sim |\mathcal{M}|$. Since $|\mathcal{M}_0| \leq |\mathcal{M}|$, and $|\mathcal{M}_0| \geq |\mathcal{M}| - (|\mathcal{M}| - |\mathcal{M}_1| + |\mathcal{M}| - |\mathcal{M}_2|) \geq |\mathcal{M}| - o(|\mathcal{M}|)$, it follows that $|\mathcal{M}_0| \sim |\mathcal{M}|$. Clearly it implies that $\log|\mathcal{M}_0| = \log|\mathcal{M}| + o(1)$, and thus inequalities (4.1), (4.2), (4.3), and (4.4) hold for $\mathcal{M}_0$ as well, which completes the proof. ∎

Remark 2. The proof of Lemma 4.2 is nonconstructive. As a result, it can prove only the existence of such a worst-case graph G_M.

We are now ready to prove Theorem 2.2.

Proof of Theorem 2.2. Let $M \in \mathcal{M}_0$ be a matrix satisfying Lemma 4.2, and consider the graph G_M, built from M. Let us show that the diameter of G_M is 2. For any two nodes x, y, denote by $\operatorname{dist}(x, y)$ the distance between x and y in G_M. The distance between any a_j (or b_j) and any v_i is at most 2 (since a_j and b_j are adjacent). The fact that $M \in \mathcal{M}_1$ implies that rows of M are pairwise noncomplementing. Thus for every i, i' there exists some j such that $m_{i,j} = m_{i',j}$, which implies $\operatorname{dist}(v_i, v_{i'}) = 2$.

Since $M \in \mathcal{M}_2$, M has the following property: for any two columns j, j' there exists some i_1, i_2, i_3, and i_4 such that $m_{i_1,j} = 1 = m_{i_1,j'}$, $m_{i_2,j} = 0 = m_{i_2,j'}$, $m_{i_3,j} = 1 \neq m_{i_3,j'}$, and $m_{i_4,j} = 0 \neq m_{i_4,j'}$. Therefore in G_M, $\operatorname{dist}(a_j, a_{j'}) = 2$, $\operatorname{dist}(b_j, b_{j'}) = 2$, $\operatorname{dist}(a_j, b_{j'}) = 2$, and $\operatorname{dist}(b_j, a_{j'}) = 2$. It follows that G_M is of diameter 2. ∎

Let $\mathcal{R} = (\mathcal{L}, \mathcal{I})$ be any interval routing scheme on G_M.

CLAIM 4.1. *For every arc (a_j, b_j) and for every vertex v_i, if $m_{i,j} = 1$ and $\mathcal{L}(v_i) \notin \mathcal{I}(a_j, b_j)$, or if $m_{i,j} = 0$ and $\mathcal{L}(v_i) \in \mathcal{I}(a_j, b_j)$, then $\mathcal{R}$ builds a path of length at least* 3.

Proof. Any "wrong" decision of $\mathcal{R}$ in routing from a_j to v_i (meaning, any decision to start the route from a_j to v_i on any outgoing arc of a_j other than the arc (a_j, b_j)) results in a route that goes through some vertex $v_{i'} \neq v_i$ before it reaches v_i. The claim now follows from the fact that there is no path shorter than two hops between any two vertices $v_i \neq v_{i'}$. ∎

Let $k = \mathrm{comp}(M)$ be the compactness of M.

CLAIM 4.2. $\mathrm{IRS}(G_M) \geq k$.

Proof. The claim is proved by showing that if there is an IRS $\mathcal{R}$ that uses no more than $k-1$ intervals per arc, then $\mathcal{R}$ builds some path of length at least 3. Since G_M is of diameter 2, this implies that $\mathcal{R}$ is not a shortest paths scheme.

Given Claim 4.1, it suffices to prove that if there is an IRS $\mathcal{R}$ that uses no more than $k-1$ intervals per arc, then $\mathcal{R}$ must make the wrong decision for some v_i and (a_j, b_j).

Let j_0 be a column of M composed of at least k blocks of consecutive ones. Such a column exists because the compactness of M is k. Let us consider the tuple $u = (u_1, \ldots, u_p)$ defined by setting $u_i = 1$ if $\mathcal{L}(v_i) \in \mathcal{I}(a_{j_0}, b_{j_0})$ and $u_i = 0$ otherwise, for every $i \in \{1, \ldots, p\}$. Since $\mathcal{I}(a_{j_0}, b_{j_0})$ is composed of at most $k-1$ intervals, u is composed of at most $k-1$ blocks of consecutive ones. Thus the column j_0 and the tuple u differ in at least one place. Let i_0 be an index such that $m_{i_0,j_0} \neq u_{i_0}$. If $u_{i_0} = 1$, then $\mathcal{L}(v_{i_0}) \in \mathcal{I}(a_{j_0}, b_{j_0})$ and $m_{i_0,j_0} = 0$. If $u_{i_0} = 0$, then $\mathcal{L}(v_{i_0}) \notin \mathcal{I}(a_{j_0}, b_{j_0})$ and $m_{i_0,j_0} = 1$. Claim 4.2 now follows by applying Claim 4.1. ∎

The order of G_M is $n = p + 2q$. Let us choose $q = \lfloor cn^{2/3} \ln^{1/3} n \rfloor$, where $c = (3/2)^{1/3} \approx 1.14$. The maximum degree of G_M is $\max\{q, \lceil p/2 \rceil + 1\} < p/2 + 2 < n/2$. (In particular, nodes a_j are connected to b_j and to the v_i's corresponding to all the 0 entries of the jth column of M.) The total number of edges in G_M is $|E| = pq + q < nq < 1.15\, n^{5/3} \ln^{1/3} n$. Replacing $p = n - 2q$ and applying Lemma 4.2, the compactness of M, k, satisfies

$$k \;\geq\; \frac{n}{4} - \frac{q}{2} - \sqrt{\frac{3(n-2q)^2}{2q} \ln(n-2q) + O\left(\frac{n^2}{q} + n \log n\right)}.$$

Noting that $O(n^2/q + n\log n) = O(n^{4/3})$ and that

$$\frac{3(n-2q)^2}{2q} \ln(n-2q) \;<\; \frac{3n^2}{2q} \ln n,$$

we get

$$\begin{aligned}
k &\geq \frac{n}{4} - \frac{c}{2} n^{2/3} \ln^{1/3} n - \sqrt{\frac{3n^2 \ln n}{2cn^{2/3} \ln^{1/3} n} + O(n^{4/3})} \\
&\geq \frac{n}{4} - \frac{c}{2} n^{2/3} \ln^{1/3} n - \sqrt{\frac{3}{2c} n^{4/3} \ln^{2/3} n} - O\left(n^{2/3}\right) \\
&\geq \frac{n}{4} - \left(\frac{c}{2} + \sqrt{\frac{3}{2c}}\right) n^{2/3} \ln^{1/3} n - O\left(n^{2/3}\right) \\
&\geq \frac{n}{4} - \left(\frac{3}{2}\right)^{4/3} n^{2/3} \ln^{1/3} n - O\left(n^{2/3}\right) \\
&> \frac{n}{4} - 1.72\, n^{2/3} \ln^{1/3} n\ .
\end{aligned}$$

Therefore, we have shown that if $\mathcal{R}$ uses at most $k-1$ intervals per arc, $\mathcal{R}$ builds a route of length at least 3. It remains to show that this result holds also if $\mathcal{R}$ uses at most $\mathrm{IRS}(G_M) - 1$ intervals per arc.

CLAIM 4.3. *For every 2-connected graph G of girth g, if $k < \text{IRS}(G)$, then the longest path of every (nonshortest path) single k-IRS is at least $\lfloor g/2 \rfloor + 1$.*

Proof. Let G be a graph as in the claim and let $\mathcal{R}$ be a single k-IRS for G. Since $\text{IRS}(G) > k$, there must exist two nodes x, y at distance d such that the routing specified by $\mathcal{R}$ from x to y is not along a shortest path. The routing path uses an alternative on a cycle between x and y. The length of this alternative path, l, satisfies $l + d \geq g$, which implies that $l \geq g/2$, because $d \leq g/2$. However, $l = g/2$ is impossible; otherwise the message would use a shortest path; hence $l > g/2 \geq \lfloor g/2 \rfloor$, and thus $l \geq \lfloor g/2 \rfloor + 1$.

Clearly, the graph G_M is 2-connected and has no triangles; thus its girth is at least 4, and therefore any single k-IRS of G_M has a routing path of length at least 3, completing the proof of Claim 4.3. ∎

Remark 3. Theorem 2.2 is tight for the length of the longest path since it is proven in [13] that $\lceil \sqrt{n \ln n} \rceil + 1$ intervals per arc are sufficient to guarantee routes of length at most $\lceil 3D/2 \rceil$, where D is the diameter of the graph. Hence for the graphs considered here, which are of diameter 2, this yields paths of length at most 3. Using this IRS, G cannot have a routing path of length 4.

To the best of our knowledge, the "best" worst-case construction which does not use randomization remains that of [9], which yields graphs G with $\text{IRS}(G) \geq n/8$, for every n power of 2.

COROLLARY 4.3. *For every sufficiently large integer n and for every integer $D \geq 2$, $D = o(n)$, there exists an n-node graph G of diameter D such that*

$$\text{IRS}(G) > \frac{n}{4} - o(n).$$

Proof. Take the worst-case n-node graph G of Theorem 2.2. G has diameter 2; therefore it has a node x of eccentricity 2. Construct a new graph G' obtained from G by adding a path of length $D-2$ to x. G' has diameter D exactly and $n' = n + D - 2$ nodes. The proof of Theorem 2.2 applies on G' as well. It turns out that G' has compactness at least $n/4 - O(n^{2/3} \log^{1/3} n)$, that is, $n'/4 - o(n')$, replacing n with $n' - D + 2 \geq n' - o(n)$. ∎

We conclude this section by showing that the lower bound can be applied to k-IRS that are not of shortest paths and not single routing schemes.

A routing scheme $\mathcal{R}$ on G is of *stretch factor s* if for all nodes x, y, $x \neq y$; the routing path length from x to y is at most s times longer than the distance in G between x and y. In particular, a shortest path k-IRS is a routing scheme of stretch factor 1.

For every integer $\alpha \geq 1$, a routing scheme $\mathcal{R}$ on G is *α-adaptive* if for all nodes x, y, $x \neq y$, there exist $\min\{\alpha, \delta\}$ edge-disjoint routing paths between x and y, where δ is the total number of "possible" edge-disjoint routing paths between x and y in G having different first edges. A single shortest path k-IRS is a 1-adaptive routing scheme of stretch factor 1. A full-adaptive k-IRS on G is a Δ-adaptive routing scheme on G, where Δ is the maximum degree of G.

Since for G_M the shortest paths between the nodes a_j and v_i are unique and since any wrong decision will route along paths of length at least 3/2 times the distance, we have the following trivial lower bound.

COROLLARY 4.4. *For every sufficiently large integer n, for every s, $1 \leq s < 3/2$, and for every integer $\alpha \geq 1$, there exists an n-node graph G such that no α-adaptive* *k-IRS of stretch factor s on G exists if*

$$k \le \frac{n}{4} - 1.72\, n^{2/3} \ln^{1/3} n.$$

5. Conclusion.

- Since the lower bound is based on the Kolmogorov complexity of the labels of the edges, the resulting bound can be applied to every kind of edge-labeling based routing scheme. Moreover, the bounds can apply to adaptive (or multipath) routing schemes.
- It would be interesting to find tighter upper bounds for small values of n, and also to express these bounds as a function of other parameters and properties of the graphs under study, such as their maximum degree, planarity, genus, tree-width, and so on.

Acknowledgment. We would like to thank Alexander Kostochka.

REFERENCES

[1] T. Eilam, S. Moran, and S. Zaks, *The complexity of the characterization of networks supporting shortest-path interval routing*, in 4th International Colloquium on Structural Information & Communication Complexity (SIROCCO), D. Krizanc and P. Widmayer, eds., Carleton Scientific, Ottawa, Canada, 1997, pp. 99–111.

[2] M. Flammini, *Deadlock-free interval routing schemes*, in 14th Annual Symposium on Theoretical Aspects of Computer Science (STACS), R. Reischuk and M. Morvan, eds., Lecture Notes in Comput. Sci., Springer, Berlin, 1997, pp. 351–362.

[3] M. Flammini, G. Gambosi, U. Nanni, and R. B. Tan, *Multidimensional interval routing schemes*, Theoret. Comput. Sci., 205 (1998), pp. 115–133.

[4] M. Flammini and G. Gambosi, *On devising Boolean routing schemes*, Theoret. Comput. Sci., 186 (1997), pp. 171–198.

[5] P. Fraigniaud and C. Gavoille, *Interval routing schemes*, Algorithmica, 21 (1998), pp. 155–182.

[6] G. N. Frederickson and R. Janardan, *Designing networks with compact routing tables*, Algorithmica, 3 (1988), pp. 171–190.

[7] C. Gavoille, *A survey on interval routing scheme*, Theoret. Comput. Sci., 245 (1999).

[8] C. Gavoille and M. Gengler, *Space-efficiency of routing schemes of stretch factor three*, in 4th International Colloquium on Structural Information & Communication Complexity (SIROCCO), D. Krizanc and P. Widmayer, eds., Carleton Scientific, Ottawa, Canada, 1997, pp. 162–175.

[9] C. Gavoille and E. Guévremont, *Worst case bounds for shortest path interval routing*, J. Algorithms, 27 (1998), pp. 1–25.

[10] C. Gavoille and S. Pérennès, *Lower bounds for interval routing on* 3*-regular networks*, in 3rd International Colloquium on Structural Information & Communication Complexity (SIROCCO), N. Santoro and P. Spirakis, Carleton University Press, Ottawa, Canada, 1996, pp. 88–103.

[11] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science*, 2nd ed., Addison-Wesley, Reading, MA, 1994.

[12] E. Kranakis, D. Krizanc, S. S. Ravi, *On multi-label linear interval routing schemes*, in 19th International Workshop on Graph-Theoretic Concepts in Computer Science—Distributed Algorithms (WG), Lecture Notes in Comput. Sci., Springer-Verlag, Berlin, 1994, pp. 338–349.

[13] R. Kráľovič, P. Ružička, and D. Štefankovič, *The complexity of shortest path and dilation bounded interval routing*, in 3rd International Euro-Par Conference, C. Lengaur, M. Griebl, and S. Gorlatch, eds., Lecture Notes in Comput. Sci., Springer, Berlin, 1997, pp. 258–265.

[14] M. Li and P. M. B. Vitányi, *An Introduction to Kolmogorov Complexity and Its Applications*, Springer-Verlag, New York, 1993.

[15] L. Narayanan and J. Opatrny, *Compact routing on chordal rings of degree four*, in 4th International Colloquium on Structural Information & Communication Complexity (SIROCCO), D. Krizanc and P. Widmayer, eds., Carleton Scientific, Ottawa, Canada, 1997, pp. 125–137.

[16] L. Narayanan and S. Shende, *Characterizations of networks supporting shortest-path interval labeling schemes*, in 3rd International Colloquium on Structural Information & Communication Complexity (SIROCCO), N. Santoro and P. Spirakis, eds., Carleton University Press, Ottawa, Canada, 1996, pp. 73–87.

[17] N. Santoro and R. Khatib, *Labelling and implicit routing in networks*, Comput. J., 28 (1985), pp. 5–8.

[18] J. van Leeuwen and R. B. Tan, *Interval routing*, Comput. J., 30 (1987), pp. 298–307.

SIAM J. DISCRETE MATH.
Vol. 12, No. 4, pp. 474–490

DETERMINANT: OLD ALGORITHMS, NEW INSIGHTS*

MEENA MAHAJAN† AND V. VINAY‡

Abstract. In this paper we approach the problem of computing the characteristic polynomial of a matrix from the combinatorial viewpoint. We present several combinatorial characterizations of the coefficients of the characteristic polynomial in terms of walks and closed walks of different kinds in the underlying graph. We develop algorithms based on these characterizations and show that they tally with well-known algorithms arrived at independently from considerations in linear algebra.

Key words. determinant, algorithms, combinatorics, graphs, matrices

AMS subject classifications. 05A15, 68R05, 05C50, 68Q25

PII. S0895480198338827

1. Introduction.

Computing the determinant, or the characteristic polynomial, of a matrix is a problem which has been studied several years ago from the numerical analysis viewpoint. In the mid 1940's, a series of algorithms which employed sequential iterative methods to compute the polynomial were proposed, the most prominent one due to Samuelson, Krylov, and Leverier [19]; see, for instance, the presentation in [10]. Then, in the 1980's, a series of parallel algorithms for the determinant were proposed by Csanky, Chistov, and Berkowitz [6, 5, 1]. This culminated in the result, shown independently by several complexity theorists including Vinay, Damm, Toda, and Valiant [26, 7, 24, 25], that computing the determinant of an integer matrix is complete for the complexity class GapL and hence computationally equivalent in a precise complexity-theoretic sense to iterated matrix multiplication or matrix powering.

In an attempt to unravel the ideas that went into designing efficient parallel algorithms for the determinant, Valiant studied Samuelson's algorithm and interpreted the computation combinatorially [25]. He presented a combinatorial theorem concerning closed walks (clows) in graphs, the correctness of which followed from that of Samuelson's algorithm. This was the first attempt to view determinant computations as graph-theoretic rather than linear algebraic manipulations. Inspired by this, and by the purely combinatorial and extremely elegant proof of the Cayley–Hamilton theorem due to Rutherford [18] (and independently discovered by Straubing [21]; see [2, 27] for nice expositions and see [3] for related material), Mahajan and Vinay [15] described a combinatorial algorithm for computing the characteristic polynomial. The proof of correctness of this algorithm is also purely combinatorial and does not rely on any linear algebra or polynomial arithmetic.

In this paper, we follow up on the work presented in [25, 21, 15] and present a unifying combinatorial framework in which to interpret and analyse a host of algorithms for computing the determinant and the characteristic polynomial. We first

*Received by the editors May 18, 1998; accepted for publication (in revised form) May 20, 1999; published electronically October 19, 1999. A preliminary version of this paper appeared in the *Proceedings of the Sixth Scandinavian Workshop on Algorithm Theory SWAT'98*, Lecture Notes in Comput. Sci. 1432, Springer-Verlag, pp. 276–287.
http://www.siam.org/journals/sidma/12-4/33882.html
†Institute of Mathematical Sciences, Chennai 600 113, India (meena@imsc.ernet.in). Part of this work was done when this author was visiting the Department of Computer Science and Automation, IISc, Bangalore, India.
‡Department of Computer Science and Automation, Indian Institute of Science, Bangalore 560 012, India (vinay@csa.iisc.ernet.in).

describe what the coefficients of the characteristic polynomial of a matrix M represent as combinatorial entities in the graph G_M whose adjacency matrix is M. We then consider various algorithms for evaluating the coefficients, and in each case we relate the intermediate steps of the computation to manipulation of similar combinatorial entities, giving combinatorial proofs of correctness of these algorithms.

In particular, in the graph-theoretic setting, computing the determinant amounts to evaluating the signed weighted sum of cycle covers. This sum involves far too many terms to allow evaluation of each, and we show how the algorithms of [19, 5, 6] essentially expand this sum to include more terms, i.e., generalizations of cycle covers, which eventually cancel out but which allow easy evaluation. The algorithm in [15] uses clow sequences explicitly; Samuelson's method [19] implicitly uses prefix clow sequences; Chistov's method [5] implicitly uses tables of tour sequences; and Csanky's algorithm [6] hinges around Leverier's lemma (see, for instance, [10]), which can be interpreted using loops and partial cycle covers. In each of these cases, we explicitly demonstrate the underlying combinatorial structures, and give proofs of correctness which are entirely combinatorial in nature.

In a sense, this paper parallels the work done by a host of combinatorialists in proving the correctness of matrix identities using the graph-theoretic setting. Foata [8] used tours and cycle covers in graphs to prove the MacMohan master theorem; Rutherford and Straubing [18, 21] reproved the Cayley–Hamilton theorem using counting over walks and cycle covers; Garsia [11], Orlin [17], and Tempereley [23] independently found combinatorial proofs of the matrix-tree theorem and Chaiken [4] generalized the proof to the all-minor matrix-tree theorem; Foata [9] and then Zeilberger [27] gave new combinatorial proofs of the Jacobi identity; Gessel [12] used transitive tournaments in graphs to prove Vandermonde's determinant identity. More recently, Minoux [16] showed an extension of the matrix-tree theorem to semirings, again using counting arguments over arborescences in graphs. For beautiful surveys of some of these results, see Zeilberger's paper [27] and chapter 4 of Stanton and White's book on constructive combinatorics [22]. Zeilberger ends with a host of "exercises" in proving many more matrix identities combinatorially.

Thus, using combinatorial interpretations and arguments to prove matrix identities has been around for a while. To our knowledge, however, a similar application of combinatorial ideas to interpret, or prove correctness of, or even develop new *algorithms* computing matrix functions, has been attempted only twice before: by Valiant [25] in 1992 and by the present authors in our earlier paper in 1997 [15]. We build on our earlier work and pursue a new thread of ideas here.

This paper is thus a collection of new interpretations and proofs of known results. The paper is by and large self-contained.

2. Matrices, determinants, and graphs. Let A be a square matrix of dimension n. For convenience, we state our results for matrices over integers, but they apply to matrices over any commutative ring.

We associate matrices of dimension n with complete directed graphs on n vertices, with weights on the edges. Let G_A denote the complete directed graph associated with the matrix A. If the vertices of G_A are numbered $\{1, 2, \ldots, n\}$, then the weight of the edge $\langle i, j \rangle$ is a_{ij}. We use the notation $[n]$ to denote the set $\{1, 2, \ldots, n\}$.

The determinant of the matrix A, $\det(A)$, is defined as the signed sum of all weighted permutations of S_n as follows:

$$\det(A) = \sum_{\sigma \in S_n} sgn(\sigma) \prod_i a_{i\sigma(i)},$$

where $sgn(\sigma) = (-1)^k$, k being the number (modulo 2) of inversions in σ, i.e., the cardinality of the set $\{\langle i, j\rangle \mid i < j, \sigma(i) > \sigma(j)\}$ modulo 2.

Each $\sigma \in S_n$ has a cycle decomposition, and it corresponds to a set of cycles in G_A. For instance, with $n = 5$, the permutation $\binom{1\,2\,3\,4\,5}{4\,2\,3\,5\,1}$ has a cycle decomposition (145)(2)(3) which corresponds to 3 cycles in G_A. Such cycles of G_A have an important property: they are all simple (nonintersecting), disjoint cycles; when put together, they touch each vertex exactly once. Such sets of cycles are called cycle covers. Note that cycle covers of G_A and permutations of S_n are in bijection with each other.

We define weights of cycle covers to correspond to weights of permutations. The weight of a cycle is the product of the weights of all edges in the cycle. The weight of a cycle cover is the product of the weights of all the cycles in it. Thus, viewing the cycle cover $\mathcal{C}$ as a set of edges, $w(\mathcal{C}) = \prod_{e \in \mathcal{C}} w(e)$. Since the weights of the edges are dictated by the matrix A, we can write $w(\mathcal{C}) = \prod_{\langle i,j\rangle \in \mathcal{C}} a_{ij}$.

We can also define the sign of a cycle cover consistent with the sign of the corresponding permutation. A cycle cover is even (resp., odd) if it contains an even number (resp., odd) of even length cycles. Equivalently, the cycle cover is even (resp., odd) if the number of cycles plus the number of edges is even (resp., odd). Define the sign of a cycle cover $\mathcal{C}$ to be $+1$ if $\mathcal{C}$ is even, and -1 if $\mathcal{C}$ is odd. Cauchy showed that with this definition, the sign of a permutation (based on inversions) and the sign of the associated cycle cover is the same. For our use, this definition of sign based on cycle covers will be more convenient.

Let $\mathcal{C}(G_A)$ denote the set of all cycle covers in the graph G_A. Then we have

$$\det(A) = \sum_{\mathcal{C} \in \mathcal{C}(G_A)} sgn(\mathcal{C}) w(\mathcal{C}) = \sum_{\mathcal{C} \in \mathcal{C}(G_A)} sgn(\mathcal{C}) \prod_{\langle i,j\rangle \in \mathcal{C}} a_{ij}.$$

Consider the characteristic polynomial of A,

$$\chi_A(\lambda) = \det(\lambda I_n - A) = c_0\lambda^n + c_1\lambda^{n-1} + \cdots + c_{n-1}\lambda + c_n.$$

To interpret these coefficients, consider the graph $G_A(\lambda)$ whose edges are labeled according to the matrix $\lambda I_n - A$. The coefficient c_l collects part of the contribution to $\det(\lambda I_n - A)$ from cycle covers having at least $(n - l)$ self-loops. (A self-loop at vertex k now carries weight $\lambda - a_{kk}$.) This is because a cycle cover with i self-loops has weight which is a polynomial of degree i in λ. For instance, with $n = 4$, consider the cycle cover $\langle 1,4\rangle, \langle 2,2\rangle, \langle 3,3\rangle, \langle 4,1\rangle$ in $G_A(\lambda)$. This has weight $(-a_{14})(\lambda - a_{22})(\lambda - a_{33})(-a_{41})$, contributing $a_{14}a_{22}a_{33}a_{41}$ to c_4, $-a_{14}a_{41}(a_{22} + a_{33})$ to c_3, $a_{14}a_{41}$ to c_2, and 0 to c_1.

Following notation from [21], we consider partial permutations, corresponding to partial cycle covers. A partial permutation σ is a permutation on a subset $S \subseteq [n]$. The set S is called the domain of σ, denoted $\mathrm{dom}(\sigma)$. The completion of σ, denoted $\hat{\sigma}$, is the permutation in S_n obtained by letting all elements outside $\mathrm{dom}(\sigma)$ be fixed points. This permutation $\hat{\sigma}$ corresponds to a cycle cover $\mathcal{C}$ in G_A, and σ corresponds to a subset of the cycles in $\mathcal{C}$. We call such a subset a partial cycle cover $\mathcal{PC}$, and we call $\mathcal{C}$ the completion of $\mathcal{PC}$. A partial cycle cover is defined to have the same parity and sign as its completion. It is easy to see that the completion need not be explicitly accounted for in the parity; a partial cycle cover $\mathcal{PC}$ is even (resp., odd) iff the number of cycles in it, plus the number of edges in it, is even (resp., odd).

Getting back to the characteristic polynomial, observe that to collect the contributions to c_l, we must look at all partial cycle covers with l edges. The $n - l$ vertices left uncovered by such a partial cycle cover $\mathcal{PC}$ are the self-loops, from whose weight the λ term has been picked up. Of the l vertices covered, self-loops, if any, contribute the $-a_{kk}$ term from their weight, not the λ term. And other edges, say $\langle i,j\rangle$ for $i \neq j$, contribute weights $-a_{ij}$. Thus the weights for $\mathcal{PC}$ evidently come from the graph G_{-A}. If we interpret weights over the graph G_A, a factor of $(-1)^l$ must be accounted for independently.

Formally, we have the following definition.

DEFINITION 2.1. *A cycle is an ordered sequence of m edges $C = \langle e_1, e_2, \ldots, e_m\rangle$, where $e_i = \langle u_i, u_{i+1}\rangle$ for $i \in [m-1]$ and $e_m = \langle u_m, u_1\rangle$ and $u_1 \leq u_i$ for $i \in [m]$ and all the u_i's are distinct. u_1 is called the head of the cycle, denoted $h(C)$. The length of the cycle is $|C| = m$, and the weight of the cycle is $w(C) = \prod_{i=1}^{m} w(e_i)$. The vertex set of the cycle is $V(C) = \{u_1, \ldots, u_m\}$.*

An l-cycle cover $\mathcal{C}$ is an ordered sequence of cycles $\mathcal{C} = \langle C_1, \ldots, C_k\rangle$ such that $V(C_i) \cap V(C_j) = \phi$ for $i \neq j$, $h(C_1) < \cdots < h(C_k)$ and $|C_1| + \cdots + |C_k| = l$.

The weight of the l-cycle cover is $wt(\mathcal{C}) = \prod_{j=1}^{k} w(C_j)$, and the sign is $sgn(\mathcal{C}) = (-1)^{l+k}$.

As a matter of convention, we call n-cycle covers simply cycle covers.

PROPOSITION 2.2. *The coefficients of $\chi_A(\lambda)$ are given by*

$$c_l = (-1)^l \sum_{\mathcal{C} \text{ is an } l\text{-cycle cover in } G_A} sgn(\mathcal{C}) wt(\mathcal{C}).$$

3. Summing over permutations efficiently.

As noted in Proposition 2.2, evaluating the determinant (or for that matter, any coefficient of the characteristic polynomial) amounts to evaluating the signed weighted sum over cycle covers (partial cycle covers of appropriate length). We consider four efficient algorithms for computing this sum. Each expands this sum to include more terms which mutually cancel out. The differences between the algorithms is essentially in the extent to which the sum is expanded.

3.1. From cycle covers to clow sequences.

Generalize the notion of a cycle and a cycle cover as follows:

A *clow* is a cycle in G_A (not necessarily simple) with the property that the minimum vertex in the cycle – called the *head* – is visited only once. An l-clow sequence is a sequence of clows where the heads of the clows are in strictly increasing order and the total number of edges (counting each edge as many times as it is used) is l.

Formally, we have the following definition.

DEFINITION 3.1. *A clow is an ordered sequence of edges $C = \langle e_1, e_2, \ldots, e_m\rangle$ such that $e_i = \langle u_i, u_{i+1}\rangle$ for $i \in [m-1]$ and $e_m = \langle u_m, u_1\rangle$ and $u_1 \neq u_j$ for $j \in \{2, \ldots, m\}$ and $u_1 = min\{u_1, \ldots, u_m\}$. The vertex u_1 is called the head of the clow and denoted $h(C)$. The length of the clow is $|C| = m$, and the weight of the clow is $w(C) = \prod_{i=1}^{m} w(e_i)$.*

An l-clow sequence $\mathcal{C}$ is an ordered sequence of clows $\mathcal{C} = \langle C_1, \ldots, C_k\rangle$ such that $h(C_1) < \cdots < h(C_k)$ and $|C_1| + \cdots + |C_k| = l$.

The weight of the l-clow sequence $\mathcal{C}$ is $wt(\mathcal{C}) = \prod_{j=1}^{k} w(C_j)$, and the sign of $\mathcal{C}$ is $sgn(\mathcal{C}) = (-1)^{l+k}$.

Note that the set of l-clow sequences properly includes the set of l-cycle covers on a graph. And the sign and weight of a cycle cover are consistent with its sign and weight when viewed as a clow sequence.

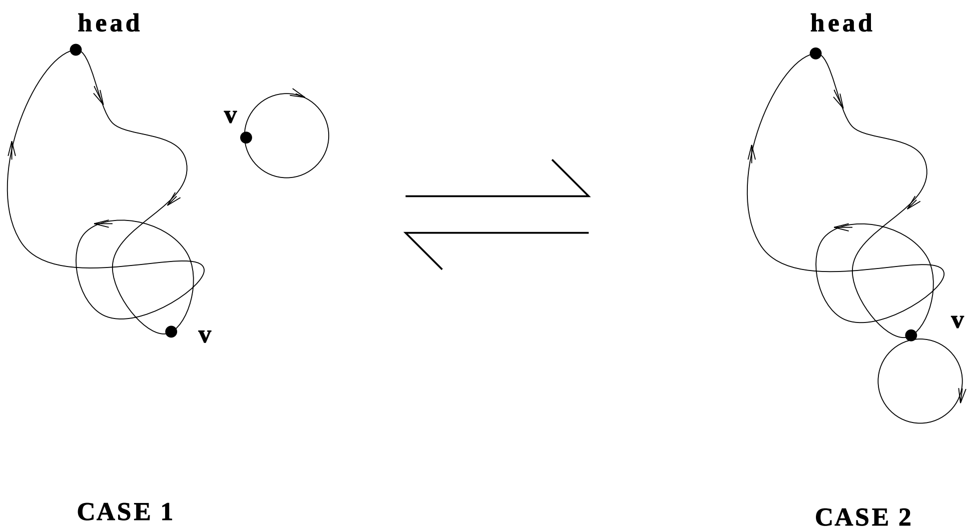


FIG. 3.1. *Pairing clow sequences of opposing signs.*

THEOREM 3.2 (see [15, Theorem 1]).

$$c_l = (-1)^l \sum_{\mathcal{C} \text{ is an } l\text{-clow sequence}} sgn(\mathcal{C}) wt(\mathcal{C}).$$

Proof. We construct an involution φ on the set of l-clow sequences. The involution has the property that φ^2 is the identity, φ maps an l-cycle cover to itself, and otherwise $\mathcal{C}$ and $\varphi(\mathcal{C})$ have the same weight but opposing signs. This shows that the contribution of l-clow sequences that are *not* l-cycle covers is zero. Consequently, only l-cycle covers contribute to the summation, yielding exactly c_l.

Let $\mathcal{C} = \langle C_1, \ldots, C_k \rangle$ be an l-clow sequence. Choose the smallest i such that C_{i+1} to C_k is a p-cycle cover for some p. If $i = 0$, the involution maps $\mathcal{C}$ to itself. Otherwise, having chosen i, traverse C_i starting from $h(C_i)$ until one of two things happen.

1. We hit a vertex that touches one of C_{i+1} to C_k.
2. We hit a vertex that completes a cycle within C_i.

Let us call the vertex v. Given the way we chose i, such a v must exist. Vertex v cannot satisfy both of the above conditions.

Case 1. Suppose v touches C_j. Map $\mathcal{C}$ to a clow sequence

$$\mathcal{C}' = \langle C_1, \ldots, C_{i-1}, C_i', C_{i+1}, \ldots, C_{j-1}, C_{j+1}, \ldots C_k \rangle.$$

The modified clow, C_i' is obtained from C_i by inserting the cycle C_j into it at the first occurence of v.

Case 2. Suppose v completes a simple cycle C in C_i. Cycle C must be disjoint from all the later cycles. We now modify the sequence $\mathcal{C}$ by deleting C from C_i and introducing C as a new clow in an appropriate position, depending on the minimum labeled vertex in C, which we make the head of C.

Figure 3.1 illustrates the mapping.

In both of the above cases, the new sequence constructed maps back to the original sequence in the opposite case. Furthermore, the number of clows in the two sequences differ by one, and hence the signs are opposing, whereas the weight is unchanged. This is the desired involution. ∎

Furthermore, the above mapping does not change the head of the first clow in the sequence. So if the goal is to compute the determinant which sums up the n-cycle covers, then the head of the first cycle must be the vertex 1. So it suffices to consider clow sequences where the first clow has head 1.

Algorithm using clow sequences. Both sequential and parallel algorithms based on the clow sequences characterization are described in [15]. We briefly describe the implementation idea below, for the case c_n.

The goal is to sum up the contribution of all clow sequences. The clow sequences can be partitioned into n groups based on the number of clows. Let C_k be the sum of the weights of all clow sequences with exactly k clows, and let $D_k = (-1)^{n+k} C_k$. Then $c_n = \sum_{k=1}^{n} D_k$.

To compute C_k, we use a divide-and-conquer approach on the number of clows: any clow sequence contributing to C_k can be suitably split into two partial clow sequences, with the left sequence having $\lceil k/2 \rceil$ clows. The heads of all clows in the left part must be less than the head of the first clow in the rightmost part. And the lengths of the left and the right partial clow sequences must add up to n. Let variable $g[p, l, u, v]$ sum up the weights of all partial clow sequences with p clows, l edges, head of first clow u, and heads of all clows at most v. (We need not consider variables where $l < p$ or $u > v$.) Then $C_k = g[k, n, 1, n]$, and such variables can be evaluated by the formula

$$g[p, l, u, v] = \begin{cases} \displaystyle\sum_{\substack{q \le r \le q + (l-p) \\ u < w \le v}} g[q, r, u, w-1] \cdot g[p-q, l-r, w, v] & \text{if } p > 1, \\ g[l, u] & \text{if } p = 1, \end{cases}$$

where $q = \lceil p/2 \rceil$. The variable $g[l, u]$ sums up the weights of all clows of length l with head u, and is also evaluated in a divide-and-conquer fashion. A clow with head u is either a self-loop if $l = 1$, or it must first visit some vertex $v > u$, find a path of length $l - 2$ to some vertex $w > u$ through vertices all greater than u, and then return to u. So

$$g[l, u] = \begin{cases} a_{uu} & \text{if } l = 1, \\ \sum_{v > u} a_{uv} \cdot a_{vu} & \text{if } l = 2, \\ \sum_{v, w > u} a_{uv} \cdot c[l-2, u, v, w] \cdot a_{wu} & \text{otherwise.} \end{cases}$$

The variable $c[l, u, v, w]$ sums the weights of all length l paths from v to w going through vertices greater than u. These variables can be evaluated as follows:

$$\begin{aligned} c[1, u, v, w] &= a_{vw} \\ c[l, u, v, w] &= \textstyle\sum_{x > u} c[p, u, v, x] \cdot c[l-p, u, x, w] \qquad \text{if } l > 1, \text{ where } p = \lceil l/2 \rceil. \end{aligned}$$

3.2. Clow sequences with the prefix property: Getting to Samuelson's method. The generalization from cycle covers to clow sequences has a certain extravagance. The reason for going to clow sequences is that evaluating their weighted sum is easy, and this sum equals the sum over cycle covers. However, there are several clow sequences which we can drop from consideration without sacrificing ease of computation. One such set arises from the following consideration:

In a cycle cover, all vertices are covered exactly once. Suppose we enumerate the vertices in the order in which they are visited in the cycle cover (following the order imposed by the cycle heads). If vertex h becomes the head of a cycle, then all vertices in this and subsequent cycles are larger than h. So all the lower numbered vertices must have been already visited. So at least $h-1$ vertices, and hence $h-1$ edges, must have been covered.

We can require our clow sequences also to satisfy this property. We formalize the prefix property: a clow sequence $\mathcal{C} = \langle C_1, \ldots, C_k \rangle$ has the prefix property if for $1 \le r \le k$, the total lengths of the clows $C_1, \ldots, C_{r-1}$ is at least $h(C_r) - 1$. A similar prefix property can be formalized for partial cycle covers. Formally, we have the following definition.

DEFINITION 3.3. *An l-clow sequence $\mathcal{C} = \langle C_1, \ldots, C_k \rangle$ is said to have the prefix property if it satisfies the following condition:*

$$\forall r \in [k], \quad \sum_{t=1}^{r-1} |C_t| \ge h(C_r) - 1 - (n-l).$$

The interesting fact is that the involution constructed in the previous subsection for clow sequences works even over this restricted set!

THEOREM 3.4 (see [25, Theorem 2]).

$$c_l = (-1)^l \sum_{\substack{\mathcal{C} \text{ is an } l\text{-clow sequence} \\ \text{with the prefix property}}} sgn(\mathcal{C}) wt(\mathcal{C}).$$

A new proof of the above theorem. In [25], Valiant observes that prefix clow sequences are the terms computed by Samuelson's method for evaluating $\chi_\lambda(A)$. Hence the correctness of the theorem follows from the correctness of Samuelson's method. And the correctness of Samuelson's method is traditionally shown using linear algebra.

Here is a simple alternative combinatorial proof of this theorem. Observe that the involution defined in the proof of Theorem 3.2 maps clow sequences with prefix property to clow sequences with prefix property. Why? Let $\mathcal{C}$ be an l-clow sequence with the prefix property satisfying case 1 in the proof. Since the length of clow C_i only increases in the process, the prefix property continues to hold. Now let $\mathcal{C}$ be an l-clow sequence with the prefix property satisfying case 2. The involution constructs a new l-clow sequence $\mathcal{C}'$ by detaching cycle C from clow C_i and inserting it later in the sequence, say between C_{j-1} and C_j. This does not change $h(C_i)$. Let $\mathcal{C}' = \mathcal{D} = \langle D_1, \ldots, D_{k+1} \rangle$; here $D_t = C_t$ for $t \in [i-1]$ or for $t = i+1$ to $j-1$, $D_i = C_i \backslash C$, $D_j = C$ and $D_{t+1} = C_t$ for $t = j$ to k. We must show that $\mathcal{D}$ has the prefix property. For $r \in [i]$, and for $r = j+1$ to $k+1$, the condition $\sum_{t=1}^{r-1} |D_t| \ge h(D_r) - 1 - (n-l)$ holds because $\mathcal{C}$ has the prefix property. Now let $i+1 \le r \le j$. Since C_i was chosen from $\mathcal{C}$ for modification, and since $i+1 \le r$, we know that $D_r, \ldots, D_{k+1}$ form a partial cycle cover, i.e., they form simple disjoint cycles. And the heads of these cycles are arranged in increasing order. So the vertices covered in $D_r, \ldots, D_{k+1}$ must all be at least as large as $h(D_r)$ and all distinct. But there are only $n - h(D_r) + 1$ such vertices. Hence

$$\begin{aligned} \textstyle\sum_{t=r}^{k+1} |D_t| &\le n - h(D_r) + 1 \\ l - \textstyle\sum_{t=1}^{r-1} |D_t| &\le n - h(D_r) + 1 \\ \textstyle\sum_{t=1}^{r-1} |D_t| &\ge h(D_r) - 1 - (n-l) \end{aligned}$$

and $\mathcal{D}$ satisfies the prefix property.

Thus in summing over all l-clow sequences with the prefix property, the only l-clow sequences that do not cancel out are the l-cycle covers, giving the claimed result. ∎

Algorithm using prefix clow sequences. To compute c_l using this characterization, we must sum up the contribution of all l-clow sequences with the prefix property. One way is to modify the dynamic programming approach used in the previous subsection for clow sequences. This can be done easily. Let us instead do things differently; the reason will become clear later.

Adopt the convention that there can be clows of length 0. Then each l-clow sequence $\mathcal{C}$ has exactly one clow C_i with head i, for $i = 1$ to n. So we write $\mathcal{C} = \langle C_1, \ldots, C_n \rangle$.

Define the signed weight of a clow C as $sw(C) = -w(C)$ if C has nonzero length, and $sw(C) = 1$ otherwise. And define the signed weight of an l-clow sequence as $sw(\mathcal{C}) = \prod_{i=1}^{n} sw(C_i)$. Then $sgn(\mathcal{C})w(\mathcal{C}) = (-1)^l sw(\mathcal{C})$. So from the preceding theorem,

$$c_l = \sum_{\substack{\mathcal{C} \text{ is an } l\text{-clow sequence} \\ \text{with the prefix property}}} sw(\mathcal{C}).$$

We say that a sequence of nonnegative integers $l_1, \ldots, l_n$ satisfies the property prefix(l) if

1. $\sum_{t=1}^{n} l_t = l$, and
2. For $r \in [n]$, $\sum_{t=1}^{r-1} l_t \geq r - 1 - (n - l)$. Alternatively $\sum_{t=r}^{n} l_t \leq n - r + 1$.

Such sequences are "allowed" as lengths of clows in the clow sequences we construct; no other sequences are allowed.

We group the clow sequences with prefix property based on the lengths of the individual clows. In a clow sequence with prefix property $\mathcal{C}$, if the length of clow C_i (the possibly empty clow with head i) is l_i, then any clow with head i and length l_i can replace C_i in $\mathcal{C}$ and still give a clow sequence satisfying the prefix property. Thus, if $z(i, p)$ denotes the total signed weight of all clows that have vertex i as head and length p, then

$$c_l = \sum_{l_1, \ldots, l_n : \text{prefix}(l)} \prod_{i=1}^{n} z(i, l_i).$$

To compute c_l efficiently, we place the values $z(i, p)$ appropriately in a series of matrices $B_1, \ldots, B_n$. The matrix B_k has entries $z(k, p)$. Since we only consider sequences satisfying prefix(l), it suffices to consider $z(k, p)$ for $p \leq n - k + 1$. Matrix B_k is of dimension $(n - k + 2) \times (n - k + 1)$ and has $z(k, p)$ on the pth lower diagonal as shown below.

$$B_k = \begin{bmatrix} z(k,0) & 0 & 0 & \cdots & 0 & 0 \\ z(k,1) & z(k,0) & 0 & \cdots & 0 & 0 \\ z(k,2) & z(k,1) & z(k,0) & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & 0 \\ z(k,n-k) & z(k,n-k-1) & z(k,n-k-2) & \cdots & z(k,1) & z(k,0) \\ z(k,n-k+1) & z(k,n-k) & z(k,n-k-1) & \cdots & z(k,2) & z(k,1) \end{bmatrix}$$

Now from the equation for c_l, it is clear that

$$c_l = \sum_{\substack{l+1=j_0 \geq j_1 \geq j_2 \geq \cdots \geq j_n = 1 : \\ j_0 - j_1, j_1 - j_2, \ldots, j_{n-1} - j_n : \text{prefix}(l)}} \left(\prod_{i=1}^{n} B_i[j_{i-1}, j_i] = \right) \quad \left(\prod_{i=1}^{n} B_i \right) [l+1, 1],$$

or more succinctly,

$$[\ c_0\ c_1\ c_2\ c_3\ \cdots\ \cdots\ c_n\ ]^T = \prod_{k=1}^{n} B_k.$$

It remains now to compute $z(i,p)$, the entries in the B matrices. We know that $z(i,0) = 1$ and $z(i,1) = -a_{ii}$. For $p \geq 2$, a clow of length p with head i must first visit a vertex $u > i$, then perform a walk of length $p-2$ via vertices greater than i to some vertex $v > i$, and then return to i. To construct the path, we exploit the fact that the (j,k)th entry in a matrix A^p gives the sum of the weights of all paths in G_A of length exactly p from j to k. So we must consider the induced subgraph with vertices $i+1, \ldots, n$. This has an adjacency matrix A_{i+1} obtained by removing the first i rows and the first i columns of A. So $A_1 = A$. Consider the submatrices of A_i as shown below.

$$A_i = \begin{pmatrix} a_{ii} & \begin{pmatrix} & R_i & \end{pmatrix} \\ \begin{pmatrix} \\ S_i \\ \\ \end{pmatrix} & \begin{pmatrix} & & \\ & A_{i+1} & \\ & & \end{pmatrix} \end{pmatrix}$$

Then the clows contributing to $z(i,p)$ must use an edge in R_i, perform a walk corresponding to A_{i+1}^{p-2}, and then return to i via an edge in S_i. In other words,

$$z(i,p) = -R_i\ A_{i+1}^{p-2}\ S_i.$$

So the matrices B_k look like this:

$$B_k = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -a_{kk} & 1 & 0 & \cdots & 0 & 0 \\ -R_k S_k & -a_{kk} & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & 0 \\ -R_k A_{k+1}^{n-k-2} S_k & -R_k A_{k+1}^{n-k-3} S_k & -R_k A_{k+1}^{n-k-4} S_k & \cdots & -a_{kk} & 1 \\ -R_k A_{k+1}^{n-k-1} S_k & -R_k A_{k+1}^{n-k-2} S_k & -R_k A_{k+1}^{n-k-3} S_1 & \cdots & -R_k S_k & -a_{kk} \end{bmatrix}$$

This method of computing $\chi_A(\lambda)$ is precisely Samuelson's method [19, 10, 1, 25]. Samuelson arrived at this formulation using Laplace's theorem on the matrix $\lambda I - A$, whereas we have arrived at it via clow sequences with the prefix property. This interpretation of the Samuelson–Berkowitz algorithm is due to Valiant [25]; the combinatorial proof of correctness (proof of Theorem 3.4) is new. (It is mentioned, without details, in [15].)

3.3. From clows to tour sequences tables: Getting to Chistov's algorithm. We now move in the other direction—generalize further beyond clow sequences. First, we relax the condition that the head of a clow may be visited only once. This gives us more generalized closed walks which we call tours. To fix a canonical representation, we do require the edges of the tour to be listed beginning from an occurrence of the head. Since there could be multiple such occurrences, we get different tours with the same multiset of edges. For instance, the tour corresponding to the vertex sequence 253246 is different from the tour corresponding to the vertex sequence 246253. Second, we deal with not just sequences but ordered lists, or tables, of sequences. Within a sequence, the tours are ordered by their heads (and all heads are distinct). However, there is no restriction on how the sequences must be ordered in the table. In fact, for the same multiset of sequences, different orderings of the sequences will give different tables that we treat as distinct. Third, the parity of a tour sequence table depends on the number of sequences in it, not the number of tours in it. A clow sequence is thus a tour sequence table where (i) each sequence contains a single tour which is a clow and (ii) within the table the sequences are ordered by their tour heads. Formally, we have the following definition.

DEFINITION 3.5. *A tour is an ordered sequence of edges $C = \langle e_1, e_2, \ldots, e_p\rangle$ such that $e_i = \langle u_i, u_{i+1}\rangle$ for $i \in [p-1]$ and $e_p = \langle u_p, u_1\rangle$ and $u_i = min\{u_1, \ldots, u_m\}$. The vertex u_1 is called the head of the tour and denoted $h(C)$. The length of the tour is $|T| = p$, and the weight of the tour is $wt(T) = \prod_{i=1}^{m} w(e_i)$.*

A j-tour sequence $\mathcal{T}$ is an ordered sequence of tours $\mathcal{T} = \langle T_1, \ldots, T_k\rangle$ such that $h(T_1) < \cdots < h(T_k)$ and $|T_1| + \cdots + |T_k| = j$. The weight of the tour sequence is $wt(\mathcal{T}) = \prod_{j=1}^{k} wt(T_j)$, and the length is $|\mathcal{T}| = j$.

An l-tour sequence table TST is an ordered sequence of tour sequences $\mathcal{F} = \langle \mathcal{T}_1, \ldots, \mathcal{T}_r\rangle$ such that $|\mathcal{T}_1| + \cdots + |\mathcal{T}_r| = l$. The weight of the TST is $wt(\mathcal{F}) = \prod_{j=1}^{r} wt(\mathcal{T}_j)$, and the sign is $(-1)^{l+r}$.

The following theorem shows that even TSTs can be used to compute the characteristic polynomial.

THEOREM 3.6.

$$c_l = (-1)^l \sum_{\mathcal{F} \text{ is an } l\text{-TST}} sgn(\mathcal{F}) wt(\mathcal{F}).$$

Proof. We present an involution on the set of l-TSTs with all l-clow sequences being fixed points, and all other l-TSTs being mapped to TSTs of the same weight but opposing sign. Since l-clow sequences which are not cycle covers also yield a net contribution of zero (see Theorem 3.2), the sum over all l-TSTs is precisely c_l.

Given an l-TST $\mathcal{F} = \langle \mathcal{T}_1, \ldots, \mathcal{T}_r\rangle$, let H be the set of all vertices which occur as heads of some tour in the table. For $S \subseteq H$, we say that S has the *clow sequence property* if the following holds: There is an $i \le r$ such that:

1. The tour sequences $\mathcal{T}_{i+1}, \ldots, \mathcal{T}_r$ are all single-tour sequences (say tour sequence $\mathcal{T}_j$ is the tour T_j).
2. No tour in any of the tour sequences $\mathcal{T}_1, \ldots, \mathcal{T}_i$ has a head vertex in S.
3. Each vertex in S is the head of a tour T_j for some $i+1 \le j \le r$., i.e., $\{h(T_j) \mid j = i+1, \ldots, r\} = S$.
4. The tour sequence table $\langle \mathcal{T}_{i+1}, \ldots, \mathcal{T}_r\rangle$ actually forms a clow sequence, i.e., the tours T_j for $i+1 \le j \le r$ are clows, and $h(T_{i+1}) < \cdots < h(T_r)$.

In other words, all tours in $\mathcal{F}$ whose heads are in S are actually clows which occur in a contiguous block of single-tour sequences, arranged in strictly increasing order of heads, and this block is not followed by any other tour sequences in $\mathcal{F}$.

Note that the empty set vacuously has the clow sequence property.

Example. In the TST $\langle\ \langle 1,2,5\rangle,\langle 3\rangle,\langle 4\rangle,\langle 6\rangle\ \rangle$, where only tour heads have been represented and where all tours are clows, $\{3,4,6\}$ has this property but $\{3,4\}$, $\{3,6\}$, $\{5,6\}$ do not.

Now, in H, find the smallest vertex v such that $H_{>v} = \{h \in H \mid h > v\}$ has the clow sequence property but $H_{\geq v} = \{h \in H \mid h \geq v\}$ does not.

If no such v exists, then H satisfies the clow sequence property, and hence $\mathcal{F}$ is an l-clow sequence. In this case, map $\mathcal{F}$ to itself.

If such a v exists, then locate the first tour sequence $\mathcal{T}_i = \langle T_1,\ldots,T_k\rangle$ where v appears (as a head). Then v is the head of the last tour T_k, because all tours with larger heads occur in a contiguous block of single-tour sequences at the end. The tour T_k can be uniquely decomposed as TC, where T is a tour and C a clow, both with head v.

Case 1. $T \neq \phi$. Map this l-TST to an l-TST where $\mathcal{T}_i$ is replaced, at the same position, by the following two tour sequences: $\langle C\rangle,\langle T_1,\ldots,T_{k-1},T\rangle$. This preserves weight but inverts the sign. In the modified l-TST, the newly introduced sequence containing only C will be chosen for modification as in Case 3.

Case 2. $T = \phi$, and $k > 1$. Map this l-TST to an l-TST where $\mathcal{T}_i$ is replaced, at the same position, by the following two tour sequences: $\langle C\rangle,\langle T_1,\ldots,T_{k-1}\rangle$. This too preserves weight but inverts the sign. In the modified l-TST, the newly introduced sequence containing only C will be chosen for modification as in Case 3.

Case 3(a). $T = \phi$ and $k = 1$. Then a tour sequence $\mathcal{T}_{i+1}$ must exist, since otherwise $H_{\geq v}$ would satisfy the clow sequence property. Now, if $\mathcal{T}_{i+1}$ has a tour with head greater than v, then, since $H_{>v}$ satisfies the clow sequence property, the TST $\mathcal{T}_{i+1},\ldots,\mathcal{T}_r$ must be a clow sequence. But recall that T has the first occurrence of v as a head and is itself a clow, so then $\mathcal{T}_i,\ldots,\mathcal{T}_r$ must also be a clow sequence, and $H_{\geq v}$ also satisfies the clow sequence property, contradicting our choice of v. Thus $\mathcal{T}_{i+1}$ must have all tours with heads at most v. Let $\mathcal{T}_{i+1} = \langle P_1,\ldots,P_s\rangle$. Now there are two subcases depending on the head of the last tour P_s.

Case 3(b). $h(P_s) = v$. Form the tour $P_s' = P_sC$. Map this l-TST to a new l-TST where the tour sequences $\mathcal{T}_i$ and $\mathcal{T}_{i+1}$ are replaced, at the same position, by a single tour sequence $\langle P_1,\ldots,P_{s-1},P_s'\rangle$. The weight is preserved and the sign inverted, and in the modified l-TST, the tour P_s' in this new tour sequence will be chosen for modification as in Case 1.

Case 3(c). $h(P_s) \neq v$. Map this l-TST to a new l-TST where the tour sequences $\mathcal{T}_i$ and $\mathcal{T}_{i+1}$ are replaced, at the same position, by a single tour sequence $\langle P_1,\ldots,P_s,C\rangle$. The weight is preserved and the sign inverted, and in the modified l-TST, the tour C in this new tour sequence will be chosen for modification as in Case 2.

Thus l-TSTs which are not l-clow sequences yield a net contribution of zero. ∎

The involution may be simpler to follow if we modify the notation as follows: decompose each tour uniquely into one or more clows with the same head and represent these clows in the same order in which they occur in the tour. Now a TST is a table of sequences of clows where, within a sequence, clows are ordered in *nondecreasing* order of head. It is easy to see that we are still talking of the same set of objects, but only representing them differently. Now, the involution picks the vertex v as above, picks the first tour sequence where v occurs as a head, picks the last clow in this sequence, and either moves this clow to a new sequence if it is not alone in its sequence, as in Cases 1 and 2, or appends it to the following sequence, as in Case 3.

Example. For a TST $\mathcal{F}$ represented using clows, let the clow heads be as shown below:

$$\langle\ \langle 1,2,2,5,5\rangle, \langle 3\rangle, \langle 4\rangle, \langle 6\rangle\ \rangle$$

Vertex 5 is chosen as v, the first tour sequence is chosen, and as dictated by Case 1, this TST is mapped to a new TST $\mathcal{F}'$ with the tours rearranged as shown below:

$$\langle\ \langle 5\rangle\langle 1,2,2,5\rangle, \langle 3\rangle, \langle 4\rangle, \langle 6\rangle\ \rangle$$

In $\mathcal{F}'$ again vertex 5 is chosen, and the first tour sequence is merged with the second as dictated by Case 3(a), to get back $\mathcal{F}$.

If the first tour sequence of $\mathcal{F}$ were $\langle 1,2,2,5\rangle$ instead, then by Case 2, $\mathcal{F}$ would be mapped to

$$\langle\ \langle 5\rangle\langle 1,2,2\rangle, \langle 3\rangle, \langle 4\rangle, \langle 6\rangle\ \rangle,$$

from which $\mathcal{F}$ would be recovered by Case 3(b).)

Algorithm using tour sequence tables. We show how grouping the l-TSTs in a carefully chosen fashion gives a formulation which is easy to compute.

Define $e_l = (-1)^l c_l$; then

$$e_l = \sum_{\mathcal{F} \text{ is an } l\text{-TST}} sgn(\mathcal{F}) wt(\mathcal{F}).$$

To compute c_l and hence e_l using this characterization, we need to compute the contributions of all l-TSTs. This is more easily achieved if we partition these contributions into l groups depending on how many edges are used up in the first tour sequence of the table. Group j contains l-TSTs of the form $\mathcal{F} = \langle \mathcal{T}_1, \ldots, \mathcal{T}_r\rangle$ where $|\mathcal{T}_1| = j$. Then $\mathcal{F}' = \langle \mathcal{T}_2, \ldots, \mathcal{T}_r\rangle$ forms an $(l-j)$-TST, and $sgn(\mathcal{F}) = -sgn(\mathcal{F}')$ and $wt(\mathcal{F}) = wt(\mathcal{T}_1)wt(\mathcal{F}')$. So the net contribution to e_l from this group, say $e_l(j)$, can be factorized as

$$\begin{aligned} e_l(j) &= \sum_{\substack{\mathcal{T}:\ j\text{-tour sequence} \\ \mathcal{F}':\ (l-j)\text{-TST}}} -sgn(\mathcal{F}')wt(\mathcal{F}')wt(\mathcal{T}) \\ &= -\left(\sum_{\mathcal{T}:\ j\text{-tour sequence}} wt(\mathcal{T})\right)\left(\sum_{\mathcal{F}':\ (l-j)\text{-TST}} sgn(\mathcal{F}')wt(\mathcal{F}')\right), \\ &= -d_j e_{l-j} \end{aligned}$$

where d_j is the sum of the weights of all j-tour sequences.

Now we need to compute d_j.

It is easy to see that $A^l[1,1]$ gives the sum of the weights of all tours of length l with head 1. To find a similar sum over tours with head k, we must consider the induced subgraph with vertices $k, k+1, \ldots, n$. This has an adjacency matrix A_k, obtained by removing the first $k-1$ rows and the first $k-1$ columns of A. (We have already exploited these properties in section 3.2.) Let $y(l,k)$ denote the sum of the weights of all l-tours with head k. Then $y(l,k) = A_k^l[1,1]$.

The weight of a j-tour sequence $\mathcal{T}$ can be split into n factors: the kth factor is 1 if $\mathcal{T}$ has no tour with head k, and is the weight of this (unique) tour otherwise. Thus

$$\begin{aligned} d_j &= \sum_{0\le l_i \le j:\ \ l_1+\cdots+l_n=j} \prod_{i=1}^n y(l_i, i) \\ &= \sum_{0\le l_i \le j:\ \ l_1+\cdots+l_n=j} \prod_{i=1}^n A_i^{l_i}[1,1]. \end{aligned}$$

Let us define a power series $D(x) = \sum_{j=0}^{\infty} d_j x^j$. Then, using the above expression for d_j, we can write

$$D(x) = \left(\sum_{l=0}^{\infty} x^l A_1^l[1,1] \right) \left(\sum_{l=0}^{\infty} x^l A_2^l[1,1] \right) \dots \left(\sum_{l=0}^{\infty} x^l A_n^l[1,1] \right).$$

Since we are interested in d_j only for $j \leq n$, we can ignore monomials of degree greater than n. This allows us to evaluate the first $n+1$ coefficients of $D(x)$ using matrix powering and polynomial arithmetic. And now e_l can be computed inductively using the following expression:

$$e_l = \sum_{j=1}^{l} e_l(j) = \sum_{j=1}^{l} -d_j e_{l-j}.$$

But this closely matches Chistov's algorithm [5]! The only difference is that Chistov started off with various algebraic entities, manipulated them using polynomial arithmetic, and derived the above formulation, whereas we started off with TSTs which are combinatorial entities, grouped them suitably, and arrived at the same formulation. And at the end, Chistov uses polynomial arithmetic to combine the computation of $D(x)$ and e_l. For completeness, we sketch below how Chistov arrived at this formulation.

Chistov's algorithm adopts the following technique (see, for example, [13]): Let C_i be the submatrix obtained by deleting the first $n-i$ rows and first $n-i$ columns of A. (In our earlier notation, C_i is the matrix A_{n-i+1}. We use C_i here to keep subscripts shorter.) Let $\Delta_i(x)$ be the determinant of $E_i = I_i - xC_i$, where I_i is the $i \times i$ identity matrix. Then $\chi_A(\lambda) = \lambda^n \Delta_n(1/\lambda)$. First, express $1/\Delta_n(x)$ as a formal power series as follows: Let $\Delta_0(x) \equiv 1$, then

$$\frac{1}{\Delta_n(x)} = \frac{\Delta_{n-1}(x)}{\Delta_n(x)} \cdot \frac{\Delta_{n-2}(x)}{\Delta_{n-1}(x)} \cdots \frac{\Delta_0(x)}{\Delta_1(x)}.$$

But $\Delta_{i-1}(x)$ and $\Delta_i(x)$ are easily related using matrix inverses:

$$\frac{\Delta_{i-1}(x)}{\Delta_i(x)} = \frac{\det(E_{i-1})}{\det(E_i)} = (E_i^{-1})[1,1].$$

Furthermore, it is easy to verify that $E_i^{-1} = (I_i - xC_i)^{-1} = \sum_{j=0}^{\infty} x^j C_i^j$. Thus,

$$\frac{1}{\Delta_n(x)} = \left(\sum_{j=0}^{\infty} x^j (C_n^j)[1,1] \right) \left(\sum_{j=0}^{\infty} x^j (C_{n-1}^j)[1,1] \right) \dots \left(\sum_{j=0}^{\infty} x^j (C_1^j)[1,1] \right).$$

Let f_j be the coefficient of x^j in $1/\Delta_n(x)$.

Now, since $\Delta_n(x) \times 1/\Delta_n(x) \equiv 1$, all coefficients other than that of the constant term must be 0. This gives us equations relating the coefficients of $\Delta_n(x)$, and hence of $\chi_A(\lambda)$, to those of $1/\Delta_n(x)$. Let $1/\Delta_n(x) = 1 - xH(x)$, where $H(x) = -(f_1 + f_2 x + f_3 x^2 + \cdots)$. Then

$$\Delta_n(x) = \sum_{i \geq 0} x^i \left[H(x)\right]^i = c_0 x^n + c_1 x^{n-1} + \cdots + c_{n-1} x + c_n.$$

So c_{n-m}, the coefficient of x^m in $\Delta_n(x)$, is given by

$$\begin{aligned} c_{n-m} &= \sum_{i\geq 0} \text{ coefficient of } x^m \text{ in } x^i\,[H(x)]^i \\ &= \sum_{i=0}^{m} \text{ coefficient of } x^{m-i} \text{ in } [H(x)]^i . \end{aligned}$$

Since only the coefficients up to x^n of any power of $H(x)$ are used, the entire computation (of $1/\Delta_n(x)$ and $\Delta_n(x)$) may be done mod x^{n+1}, giving an NC algorithm.

Note that the expression for $1/\Delta_n(x)$ obtained above is precisely the power series $D(x)$ we defined to compute the contributions of j-tour sequences.

3.4. Relating tours and cycle covers: Getting to Csanky's algorithm. We now consider the most unstructured generalization of a cycle: we relax the condition that a tour must begin from an occurrence of the minimum vertex. All we are interested in is a closed path, and we call such paths loops. Formally, we have the following definition.

DEFINITION 3.7. *A loop at vertex v is a walk from v to v; i.e., a loop L is an ordered sequence of edges $L = \langle e_1, e_2, \ldots, e_p\rangle$ such that $e_i = \langle u_i, u_{i+1}\rangle$ for $i \in [p-1]$ and $u_{p+1} = u_1$. The loop has length p and weight $\prod_{i=1}^{p} w(e_i)$.*

Having relaxed the structure of a loop, we now severely limit the way in which loops can be combined in sequences. A loop may be combined only with a partial cycle cover. Similar in spirit to Theorems 3.2, 3.4, and 3.6, we now show cancellations among such combinations.

THEOREM 3.8. *For $k \in \{1, \ldots, n\}$,*

$$kc_k + \sum_{j=1}^{k} c_{k-j}\left(\sum_{L \text{ is a loop of length } j \text{ in } G_A} w(L)\right) = 0.$$

It is easy to see that $A^j[i,i]$ sums the weights of all paths of length j from i to i in G_A. Such paths are loops; thus, $\sum_{i=1}^{n} A^j[i,i]$ sums the weights of all loops of length j in G_A. But $\sum_{i=1}^{n} A^j[i,i] = s_j$, the trace of the matrix A^j. Thus the above theorem is merely Leverier's lemma, usually stated as follows.

LEMMA 3.9 (Leverier's lemma [10, 13]). *The coefficients of the characteristic polynomial of a matrix A satisfy the following equalities:*

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ s_1 & 2 & 0 & \cdots & 0 & 0 \\ s_2 & s_1 & 3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & 0 \\ s_{n-2} & s_{n-3} & s_{n-4} & \cdots & n-1 & 0 \\ s_{n-1} & s_{n-2} & s_{n-3} & \cdots & s_1 & n \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ \vdots \\ c_n \end{bmatrix} = - \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ \vdots \\ s_n \end{bmatrix},$$

where s_j is the trace of the matrix A_j.

A combinatorial proof of the above lemma. Consider the kth claimed equality,

$$kc_k + \sum_{j=1}^{k} s_j c_{k-j} = 0,$$

where $c_0 = 1$. The terms contributing to $\sum_{j=1}^{k} s_j c_{k-j}$ consist of loops of length j and partial cycle covers of length $k-j$. The loops carry only a weight but no sign, whereas the partial cycle covers are weighted and signed. We show how to achieve cancellations within this set.

Let S be a loop of length j, and let $\mathcal{C}$ be a $(k-j)$-cycle cover.

Case 1. S forms a simple cycle, disjoint from all the cycles in $\mathcal{C}$. In this case, S can be merged into C to form a k-cycle cover $\mathcal{C}'$, with weight $wt(S)wt(\mathcal{C})$ and sign $-sgn(\mathcal{C})$. This will cancel against one copy of $\mathcal{C}'$ coming from the kc_k part. What about the $k-1$ remaining copies? Note that if $\mathcal{C}' = \langle C_1, \ldots, C_l \rangle$, then each C_i can be pulled out to give a partition into a loop S and a cycle cover C, cancelling against the corresponding term from $s_{l_i} c_{k-l_i}$. Furthermore, each C_i can be written as a loop in l_i different ways, depending on the starting point. So $\mathcal{C}'$ gives rise to $\sum_{i=1}^{l} |l_i| = k$ pairs of the form (loop, partial cycle cover); hence the term kc_k is accounted for.

Case 2. S and $\mathcal{C}$ cannot be merged into a k-cycle cover. Start traversing the loop S until one of the following things happen:

1. S hits a vertex v in $\mathcal{C}$.
2. S revisits a vertex v.

Only one of the two can happen first. Suppose S touches cycle C of $\mathcal{C}$. Let $|C| = l$. Consider the new pair $S', \mathcal{C}'$, where cycle C is removed from $\mathcal{C}$ and inserted in S at the first possible position v. This pair contributes the same weight with opposite sign to the term $s_{j+l} c_{k-j-l}$, and these two terms cancel out. Now suppose S completes a simple cycle C of length l within itself without touching $\mathcal{C}$. Consider the new pair $S', \mathcal{C}'$, where cycle C is removed from S and inserted in $\mathcal{C}$ at the appropriate position. This pair contributes the same weight with opposite sign to the term $s_{j-l} c_{k-j+l}$, where $|C| = l$, and these two terms cancel out. □

Algorithm using loops. Any algorithm for computing $\chi_A(\lambda)$ that uses Leverier's lemma implicitly exploits cancellations among loops and partial cycle covers in computing a sum which evaluates to precisely l-cycle covers. A straightforward sequential algorithm is to first compute for each j, the sum s_j, of the weights of all loops of length j in G_A using either matrix powering or dynamic programming, and then to compute $c_1, c_2, \ldots, c_n$ in order using the recurrence. Csanky's implementation [6] directly uses matrix inversion to compute the c_j's in parallel from the values of s_i's.

Note that l-loops are closed paths of length l with *no* restriction on the ordering of the edges. As sequences of edges, they thus subsume tours, clows, and cycles. In this sense, Csanky's algorithm is more extravagant than the others described above. On the other hand, it is the most frugal in allowing combinations; a loop may only be combined with a partial cycle cover and not with other loops.

4. Discussion. Starting with the definition of the coefficients of the characteristic polynomial as the signed weighted sum of all partial cycle covers, we have considered several ways of expanding the summation while keeping the net contribution the same. In a sense, the expansion corresponding to clow sequences with the prefix property, as generated by Samuelson's method, is the most conservative. All the other expansions we have considered include these sequences and more. (A clow is a tour is a loop, but not vice versa.) There are smaller expansions that still cancel out nicely (for instance, consider clow sequences where $C_i \cap C_j \neq \phi$ for at most one pair i, j). However, these smaller expansions do not seem to yield efficient computational methods. Can this observation be formally proved, i.e., can one show that any efficient method for computing c_l must include at least the l-clow sequences with the prefix property?

One of the oldest methods for computing the determinant is Gaussian elimination. Strassen ([20] or see the textbook presentation in [14]) shows how to obtain a division-free code corresponding to Gaussian elimination. Can this method also be interpreted combinatorially?

If we assume that addition, subtraction, multiplication, and division are unit-cost operations, then Gaussian elimination remains one of the most efficent methods for computing the determinant. Can this efficiency be explained combinatorially? Strassen's interpretation uses formal power series expansions and shows how Gaussian elimination uses the entire power series rather than a truncated polynomial. So high degree monomials are generated, corresponding to sequences of clows or tours or loops of arbitrary length, not just restricted to n. Is this where the computational advantage lies — do higher degrees help?

Acknowledgments. We thank Ashok Subramanian for showing how our combinatorial interpretations and construction in the clow sequences approach extend to Chistov's algorithm as well. We thank an anonymous referee for bringing the references [2, 3, 18] to our notice and for providing several comments that have improved the readability of this paper.

REFERENCES

[1] 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,

[2] R. A. Brualdi, *The many facets of combinatorial matrix theory*, in Matrix Theory and Applications, C. R. Johnson, ed., AMS, Providence, RI, 1990, pp. 1–35.

[3] R. A. Brualdi and H. Ryser, *Combinatorial Matrix Theory*, Cambridge University Press, Cambridge, UK, 1991.

[4] S. Chaiken, *A combinatorial proof of the all minors matrix theorem*, SIAM J. Alg. Discrete Methods, 3 (1982), pp. 319–329.

[5] A. L. Chistov, *Fast parallel calculation of the rank of matrices over a field of arbitrary characteristic*, in Proc. Int. Conf. Foundations of Computation Theory, Lecture Notes in Comput. Sci. 199, Springer-Verlag, New York, Berlin, 1985, pp. 63–69.

[6] L. Csanky, *Fast parallel inversion algorithm*, SIAM J. Comput., 5 (1976), pp. 618–623.

[7] C. Damm, DET=$L^{(\#L)}$, preprint 8 Fachbereich Informatik der Humboldt–Universität zu Berlin, Informatik, 1991.

[8] D. Foata, *Etude algébrique de certains problèmes d'analyse combinatoire et du calcul des probabilités*, Publ. Inst. Statist. Univ. Paris, 14 (1965), pp. 81–241.

[9] D. Foata, *A combinatorial proof of Jacobi's identity*, Ann. Discrete Math., 6 (1980), pp. 125–135.

[10] D. Fadeev and V. Fadeeva *Computational Methods in Linear Algebra*, Freeman, San Francisco, 1963.

[11] A. Garsia and Ö. Eğecioğlu, *Combinatorial Foundations of Computer Science*, unpublished collection.

[12] I. Gessel, *Tournaments and Vandermonde's determinant*, J. Graph Theory, 3 (1979), pp. 305–307.

[13] D. Kozen, *The Design and Analysis of Algorithms*, Springer-Verlag, New York, 1992.

[14] T. F. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann Publishers Inc., San Mateo, 1992.

[15] M. Mahajan and V. Vinay, *Determinant: combinatorics, algorithms, complexity*, Chicago J. Theoret. Comput. Sci., 1997, pp. 5.

[16] M. Minoux, *Bideterminants, arborescences and extension of the matrix-tree theorem to semirings*, Discrete Math., 171 (1997), pp. 191–200.

[17] J. B. Orlin, *Line-digraphs, arborescences, and theorems of Tutte and Knuth*, J. Combin. Theory Ser. B, 25 (1978) pp. 187–198.

[18] D. E. Rutherford, *The Cayley-Hamilton theorem for semirings*, Proc. Roy. Soc. Edinburgh Sect. A, 66 (1964), pp. 211–215.

[19] P. A. Samuelson, *A method of determining explicitly the coefficients of the characteristic polynomial*, Ann. Math. Stat., 13 (1942) pp. 424–429.

[20] V. Strassen, *Vermeidung von divisionen*, J. Reine Angew. Math., 264 (1973), pp. 182–202.

[21] H. Straubing, *A combinatorial proof of the Cayley-Hamilton theorem*, Discrete Math., 43 (1983), pp. 273–279.

[22] D. Stanton and D. White, *Constructive Combinatorics*, Springer-Verlag, New York, Berlin, 1986.

[23] H. N. V. Tempereley, *Graph Theory and Applications*, Ellis Horwood, Chichester, 1981.

[24] S. Toda, *Counting Problems Computationally Equivalent to the Determinant*, manuscript, 1991

[25] L. G. Valiant, *Why is Boolean complexity theory difficult?*, in Boolean Function Complexity, M. S. Paterson, ed., London Math. Soc. Lecture Notes Ser. 169, Cambridge University Press, Cambridge, UK, 1992, pp. 84–94.

[26] V. Vinay, *Counting auxiliary pushdown automata and semi-unbounded arithmetic circuits*, in Proceedings of the 6th Annual Conference on Structure in Complexity Theory, Chicago, IL, 1991, pp. 270–284.

[27] D. Zeilberger, *A combinatorial approach to matrix algebra*, Discrete Math., 56 (1985), pp. 61–72.

SIAM J. Discrete Math.
Vol. 12, No. 4, pp. 491–499

STAR EXTREMAL CIRCULANT GRAPHS*

KO-WEI LIH[†], DAPHNE DER-FEN LIU[‡], AND XUDING ZHU[§]

Abstract. A graph is called star extremal if its fractional chromatic number is equal to its circular chromatic number (also known as the star chromatic number). We prove that members of a certain family of circulant graphs are star extremal. The result generalizes some known theorems of Sidorenko [*Discrete Math.*, 91 (1991), pp. 215–217] and Gao and Zhu [*Discrete Math.*, 152 (1996), pp. 147–156]. We show relations between circulant graphs and distance graphs and discuss their star extremality. Furthermore, we give counterexamples to two conjectures of Collins [*SIAM J. Discrete Math.*, 11 (1998), pp. 330–339] on asymptotic independence ratios of circulant graphs.

Key words. circular chromatic number, fractional chromatic number, circulant graph, distance graph, star extremal graph, independence ratio

AMS subject classification. 05C15

PII. S0895480198342838

1. Introduction.

Given a positive integer n and a set $S \subseteq \{1, 2, 3, \ldots, \lfloor n/2 \rfloor\}$, let $G(n, S)$ denote the graph with vertex set $V(G) = \{0, 1, 2, \ldots, n-1\}$ and edge set $E(G) = \{uv : |u - v|_n \in S\}$, where $|x|_n := \min\{|x|, n - |x|\}$ is the *circular distance modulo n*. Then $G(n, S)$ is called the *circulant graph* of order n with the *generating set S*.

Circulant graphs have been investigated in different fields. Such graphs are "star polygons" to geometers [7]. The well known Ádám's conjecture [1] states: $G(n, S)$ and $G(n, S')$ are isomorphic if and only if $S' = kS = \{ks : s \in S\}$ for some unity k in the ring Z_n. Alspach and Parsons [2] proved that this conjecture does not hold in general. However, it is true for some special classes of circulant graphs. Parsons [15] characterized the set A_k of connected circulant graphs $G(n, S)$ such that the neighbors $N(x)$ for each vertex x induce a k-cycle in $G(n, S)$. Then Ádám's conjecture was established for circulant graphs in A_k.

In this article, we explore the star extremality of circulant graphs. A graph is called *star extremal* if its fractional chromatic number and circular chromatic number, defined below, are equal.

A *fractional coloring* of a graph G is a mapping c from $\mathcal{I}(G)$, the set of all independent sets of G, to the interval $[0, 1]$ of real numbers such that $\sum \{c(I) : x \in I \text{ and } I \in \mathcal{I}(G)\} \geq 1$ for any vertex x in G. The *fractional chromatic number* $\chi_f(G)$

*Received by the editors August 3, 1998; accepted for publication (in revised form) May 28, 1999; published electronically October 19, 1999. An extended abstract of this paper appear in Combinatorics, Computation and Logic, Proceedings of DMTCS '99 and CATS '99, Auckland, New Zealand, January 18-21, 1999, Australian Computer Science Communications, Vol. 21, No. 3, C. S. Calude and M. J. Dinneen, eds., Springer-Verlag, Singapore, 1999, pp. 301–309.

http://www.siam.org/journals/sidma/12-4/34283.html

[†]Institute of Mathematics, Academia Sinica, Nankang, Taipei, Taiwan 11529 (makwlih@sinica.edu.tw). The research of this author was supported in part by National Science Council of the R.O.C. grant NSC87-2115-M-001-004.

[‡]Department of Mathematics and Computer Science, California State University, Los Angeles, CA 90032 (dliu@calstatela.edu). The research of this author was supported in part by National Science Foundation grant DMS-9805945.

[§]Department of Applied Mathematics, National Sun Yat-sen University, Kaoshiung, Taiwan 80424 (zhu@math.nsysu.edu.tw). The research of this author was supported in part by National Science Council of R.O.C. grant NSC88-2115-M-110-001.

of G is the infimum of the *weight*, $w(c) = \sum \{c(I) : I \in \mathcal{I}(G)\}$, of a fractional coloring c of G. For a different but equivalent definition of the fractional chromatic number, we refer the reader to [17].

Let k and d be positive integers such that $k \geq 2d$. A (k,d)-coloring of a graph $G = (V,E)$ is a mapping c from V to $\{0,1,\ldots,k-1\}$ such that $|c(x) - c(y)|_k \geq d$ for any edge xy in G. The *circular chromatic number* $\chi_c(G)$ of G is the infimum of k/d for which there exists a (k,d)-coloring of G. The circular chromatic number is also known as the *star chromatic number* in the literature [20, 21].

For any graph G, it is well known [22] that

$$(*) \qquad \max\left\{\omega(G), \frac{|V(G)|}{\alpha(G)}\right\} \leq \chi_f(G) \leq \chi_c(G) \leq \chi(G); \ \lceil \chi_c(G) \rceil = \chi(G),$$

where $\omega(G)$ is the *clique number* (i.e., the maximum number of vertices of a complete subgraph in G); $\alpha(G)$ is the *independence number* (i.e., the maximum number of vertices of an independent set in G). Hence, a graph G is star extremal if the equality holds in the second inequality in $(*)$.

The notion of star extremality for graphs arose from the study of the chromatic number and the circular chromatic number of the lexicographic product of graphs. The lexicographic product $G[H]$ of G and H is the graph with vertex set $V(G)\times V(H)$ and in which $(v_1,w_1)(v_2,w_2)$ is an edge if and only if $v_1v_2 \in E(G)$ or $v_1 = v_2$ and $w_1w_2 \in E(H)$. (Informally, we substitute a copy of H for each vertex of G.) It was proved in [10] that, if G is star extremal, then $\chi_c(G[H]) = \chi_c(G)\chi(H)$ for any graph H. Therefore for any star extremal graph G, the circular chromatic number, and hence the chromatic number, of the lexicographic product $G[H]$ is determined by $\chi_c(G)$ and $\chi(H)$. Klavžar [12] also used star extremal graphs to investigate the chromatic numbers of lexicographic products of graphs.

The star extremality for circulant graphs was first discussed by Gao and Zhu [10]. They proved that if all the vertices of $G(n,S)$ have degree ≤ 3, then $G(n,S)$ is star extremal. On the other hand, there exist star extremal circulant graphs (see section 3 below or [10]). In general, it seems a difficult problem to determine whether or not an arbitrary circulant graph is star extremal.

As circulant graphs are vertex-transitive, we know that

$$(**) \qquad \chi_f(n,S) = n/\alpha(n,S),$$

where $\chi_f(n,S)$ and $\alpha(n,S)$ denote, respectively, the fractional chromatic number and the independence number for $G(n,S)$. Therefore, the determination of the independence number of a circulant graph is equivalent to the determination of its fractional chromatic number. Recently, Codenotti, Gerace, and Vigna [5] have proved that it is NP-complete to compute the independence number for general circulant graphs. However, for some circulant graphs, including the ones discussed in this article, the independence number can be computed in polynomial time.

In section 2, we focus on the family of circulant graphs whose generating set S consists of consecutive integers. Given integers $k < k' \leq n/2$, let $S_{k,k'}$ denote the set $\{k,(k+1),\ldots,k'\}$. We will determine the exact value of $\alpha(n,S_{k,k'})$ for any $n \geq 2k'$ and $k' \geq (5/4)k$. This result is used to prove that the circulant graphs $G(n,S_{k,k'})$ are star extremal for all $n \geq 2k'$ and $k' \geq (5/4)k$.

For some special values of k, k', and n, the circulant graphs $G(n,S_{k,k'})$ have appeared in several articles. It was proved in [10] that $G(n,S_{1,k'})$ (i.e., $k = 1$) is star extremal for all $n \geq 2k'$. When $k' = 2k-1$ and $n \geq 4k-1$, the circulant graph

$G(n, S_{k,2k-1})$ is a triangle-free regular graph with degree $2k$. Sidorenko [18] proved that $\alpha(n, S_{k,2k-1}) = 2k$ for $6k-2 \le n \le 8k-3$ and applied this result to answer a question of Erdös [16], namely, the existence of triangle-free regular graphs on $n(\ne 3, 7, 9)$ vertices with its independence number equal to the degree. Gao and Zhu [10] then applied Sidorenko's result to show that the circulant graphs $G(n, S_{k,2k-1})$ are star extremal for $6k-2 \le n \le 8k-3$.

Let Z denote the set of all integers. For a given finite set S of positive integers, the *distance graph*, denoted by $G(Z, S)$, has Z as its vertex set, and uv forms an edge if $|u-v| \in S$. Thus, the distance graph $G(Z, S)$ can be viewed as the limit of circulant graphs $G(n, S)$ as n approaches infinity. In section 3, we explore the relation of star extremality between circulant graphs and distance graphs for general sets S.

The *independence ratio* of a graph G is defined to be $\alpha(G)/|V(G)|$. In section 4, we show that for a given S, the fractional chromatic number of the distance graph $G(Z, S)$ is equal to the reciprocal of the asymptotic independence ratio of circulant graphs $G(n, S)$ as n approaches infinity. Applying this fact, we present counterexamples to two conjectures of Collins [6] on the asymptotic independence ratio of circulant graphs.

2. Circulant graphs with interval generating sets. We shall discuss the star extremality of circulant graphs, whose generating sets are of the form $S_{k,k'} = \{k, (k+1), \ldots, k'\}$, where $k < k' \le n/2$. For the case that $k' \ge (5/4)k$, we determine the exact values of $\alpha(n, S_{k,k'})$ for all n. Using this result, we show that such circulant graphs are star extremal.

One of the tools we shall use is the following *multiplier method*, which was first used in [10] and has been applied to solve problems concerning coloring of circulant graphs as well as distance graphs [3, 8, 19]. Given a circulant graph $G(n, S)$ and a positive integer t, let

$$\lambda_t(n, S) := \min\{|ti|_n : i \in S\},$$

and let

$$\lambda(n, S) := \max\{\lambda_t(n, S) : t = 1, 2, 3, \ldots\},$$

where the multiplications ti are carried out modulo n and $|x|_n$ is the circular distance modulo n. For any positive integer t, the mapping c on $\{0, 1, 2, \ldots, n-1\}$ defined by $c(i) = ti$ is an $(n, \lambda_t(n, S))$-coloring for $G(n, S)$ (multiplications are carried out modulo n.) Hence, $\chi_c(n, S) \le n/\lambda(n, S)$. Combining this with $(*)$ and $(**)$, we obtain the following result.

LEMMA 2.1 (see [10]). *Let $G(n, S)$ be a circulant graph. Then $\lambda(n, S) \le \alpha(n, S)$. Moreover, if $\lambda(n, S) = \alpha(n, S)$, then $\chi_f(n, S) = \chi_c(n, S) = n/\alpha(n, S)$, i.e., $G(n, S)$ is star extremal.*

The value of $\lambda(n, S)$ can be calculated in polynomial time. To be precise, we have the following lemma.

LEMMA 2.2. *Let $G(n, S)$ be a circulant graph. Then $\lambda(n, S) = \lambda_t(n, S) = |ts|_n = |t(-s)|_n$ for some t, $1 \le t \le \lceil n/2 \rceil$, and $s \in S$.*

Proof. By definition, $\lambda_t(n, S) = \lambda_{n-t}(n, S) = \lambda_{t'}(n, S)$ for any $t \equiv t' \pmod{n}$, and $|ts|_n = |t(-s)|_n$. ∎

Sidorenko [18] proved that $\alpha(n, S_{k,2k-1}) = 2k$ if $6k-2 \le n \le 8k-3$. Later on, Gao and Zhu [10] proved that $\lambda(n, S_{k,2k-1}) = 2k$ under the same condition on n. By combining these two results with Lemma 2.1, the following was obtained in [10].

THEOREM 2.3. *If $k' = 2k-1$, then the circulant graphs $G(n, S_{k,k'})$ are star extremal for all n, where $6k-2 \le n \le 8k-3$.*

Other special subfamilies of the circulant graphs $G(n, S_{k,k'})$ that have been studied include the following two.

THEOREM 2.4 (see [10]). *If $k' \le n/2$, then $G(n, S_{1,k'})$ is star extremal and $\chi_f(n, S_{1,k'}) = \chi_c(n, S_{1,k'}) = n/\lfloor \frac{n}{k'+1} \rfloor$.*

THEOREM 2.5 (see [10]). *Suppose $k' = k+l \le n/2$. If $n - 2k' < \min\{k, l\}$, then $G(n, S_{k,k'})$ is star extremal and $\chi_f(n, S_{k,k'}) = \chi_c(n, S_{k,k'}) = n/k$.*

The proofs of Theorems 2.4 and 2.5 are obtained, respectively, by showing $\alpha(n, S_{1,k'}) = \lambda(n, S_{1,k'}) = \lfloor \frac{n}{k'+1} \rfloor$ and $\alpha(n, S_{k,k'}) = \lambda(n, S_{k,k'}) = k$ under the assumptions on k and k'.

We note here that the circulant graphs $G(n, S_{1,k'})$ in Theorem 2.4 are indeed powers of the cycle C_n on n vertices. Given n and r, the rth *power* of C_n, denoted by C_n^r, has the same vertex set as C_n, and u, v are adjacent if their distance on the cycle C_n is not greater than r. Therefore $G(n, S_{1,k'}) = C_n^{k'}$ by definition.

In their study of the circular chromatic number of planar graphs Gao, Wang, and Zhou [9] defined a family of planar graphs Q_n, called *triangular prisms*, which have vertex set $V = \{u_0, u_1, u_2, \ldots, u_{n-1}\} \cup \{v_0, v_1, v_2, \ldots, v_{n-1}\}$ and edge set E consisting of two n-cycles $(u_0, u_1, \ldots, u_{n-1})$ and $(v_0, v_1, \ldots, v_{n-1})$ and $2n$ edges (u_i, v_i), (u_{i+1}, v_i) for every $0 \le i \le n-1$ ($u_0 = u_n$). In [9], the argument for computing the values of $\chi_f(Q_n)$ is long. The proof can be shortened considerably by applying known results in circulant graphs. The family of planar graphs Q_n are precisely the second powers of even cycles. Indeed, $(v_0, u_1, v_1, u_2, v_2, \ldots, u_{n-1}, v_{n-1}, u_0)$ is a cycle of length $2n$, and $Q_n \cong C_{2n}^2$. Hence, by Theorem 2.4, $\chi_f(Q_n) = \chi_c(Q_n) = 2n/\lfloor \frac{2n}{k+1} \rfloor$.

Now we consider the general family of circulant graphs $G(n, S_{k,k'})$. We view the vertices of $G(n, S_{k,k'})$ as circularly ordered in the clockwise direction and denote by $[a, b]$ the set of integers $\{a, a+1, a+2, \ldots, b\}$, where the addition is taken under modulo n. For example, $[2, 5] = \{2, 3, 4, 5\}$ and $[5, 2] = \{5, 6, \ldots, n-1, 0, 1, 2\}$.

LEMMA 2.6. *Suppose I is an independent set of $G(n, S_{k,k'})$. Then for any j the cardinality of $I \cap [j, j+k+k'-1]$ is at most k.*

Proof. By symmetry, it suffices to show that for any independent set I, the cardinality of $I \cap [0, k+k'-1]$ is at most k. Suppose $i \in [0, k+k'-1]$ is the least element of I. Then $i+k, i+k+1, \ldots, i+k' \notin I$. Let $A = [i+1, i+k-1] \cap I$ and $B = [i+k'+1, k'+k-1] \cap I$. If $x \in A$, then $x + k' \notin B$. This implies $|A| + |B| \le k-1$. Therefore $|I \cap [0, k+k'-1]| \le k$. ◻

LEMMA 2.7. *Suppose $G = G(n, S_{k,k'})$ with $n = q(k+k') + r$, $0 \le r \le k+k'-1$. Then*

$$\lambda(G) \ge \begin{cases} \lambda_q(G) = qk & \text{if } 0 \le r \le k'; \\ \lambda_{q+1}(G) = qk + r - k' & \text{if } k'+1 \le r \le k'+k-1. \end{cases}$$

Proof. It suffices to show that $\lambda_q(G) = qk$ when $0 \le r \le k'$ and $\lambda_{q+1}(G) = qk + r - k'$ when $k'+1 \le r \le k'+k-1$.

If $0 \le r \le k'$, then

$$\lambda_q(G) = \min\{qk, q(k+1), q(k+2), \ldots, qk', \\ n - qk, n - q(k+1), n - q(k+2), \ldots, n - qk'\}.$$

Because $qk \le q(k+1) \le \cdots \le qk'$ and $n - qk \ge n - q(k+1) \ge n - q(k+2) \ge \cdots \ge n - qk'$, it is enough to show $n - qk' \ge qk$. This is true since $n - qk' = qk + r \ge qk$.

If $k' + 1 \le r \le k' + k - 1$, then

$$\lambda_{q+1}(G) = \min\{(q+1)k, (q+1)(k+1), (q+1)(k+2), \dots, (q+1)k', \\ n-(q+1)k, n-(q+1)(k+1), \dots, n-(q+1)k' \}.$$

Because $(q+1)k \le (q+1)(k+1) \le (q+1)(k+2) \le \cdots \le (q+1)k'$ and $n-(q+1)k \ge n-(q+1)(k+1) \ge n-(q+1)(k+2) \ge \cdots \ge n-(q+1)k' = qk+r-k'$, it is enough to show $qk+r-k' \le (q+1)k-1$. This is true since $qk+r-k' \le qk+(k+k'-1)-k' = (q+1)k-1$. ∎

THEOREM 2.8. *Suppose* $G = G(n, S_{k,k'})$ *and* $k' \ge (5/4)k$. *Let* $n = q(k+k')+r$, *where* $0 \le r \le k+k'-1$. *Then*

$$\alpha(G) = \lambda(G) = \begin{cases} qk & \text{if } 0 \le r \le k'; \\ qk+r-k' & \text{if } k'+1 \le r \le k'+k-1. \end{cases}$$

Equivalently, $\alpha(G) = \lambda(G) = qk + \max\{0, r-k'\}$.

Proof. Let $n = q(k+k')+r$, $0 \le r \le k+k'-1$. By Lemmas 2.1 and 2.7, it suffices to show that $\alpha(G) \le qk + \max\{0, r-k'\}$. If $q = 0$, the result follows from Lemma 2.6. Thus we may assume $q \ge 1$.

Assume to the contrary that $\alpha(G) > qk + \max\{0, r-k'\}$. Let I be a maximum independent set of G. Regard I as a disjoint union of I-intervals, where an *I-interval* is a maximal interval $[a,b]$ consisting of vertices in I. Then the length (namely, the number of vertices) of any I-interval is between 1 and k. By Lemma 2.6 and the assumption that $q \ge 1$, there are at least two I-intervals. Assume that the independent set I chosen has the minimum number of I-intervals among all maximum independent sets of G.

Two I-intervals $[a,b]$ and $[c,d]$ are called *consecutive* if $[b+1, c-1] \cap I = \emptyset$. Note that the consecutive "relation" is not symmetric, i.e., $[a,b]$ and $[c,d]$ being consecutive does not imply that $[c,d]$ and $[a,b]$ are consecutive. (Indeed, $[c,d]$ and $[a,b]$ are not consecutive if $[a,b]$ and $[c,d]$ are consecutive and I contains more than two I-intervals.) For two consecutive I-intervals $[a,b]$ and $[c,d]$, the cardinality of the set $[b+1, c-1]$ is called the *gap* between them.

First we show that if $[a,b]$ and $[c,d]$ are two consecutive I-intervals, then $b+k'+1 \le c+k-1$ (or equivalently, $b-k+1 \le c-k'-1$). Here we assume, without loss of generality, that $0 \le a \le b < c \le d \le n-1$. Suppose to the contrary that $b+k'+1 \ge c+k$. For any $b+1 \le x \le c-1$, if y is adjacent to x then straightforward calculations show that y is adjacent to either b or c. Hence none of the neighbors of x is in I. This implies that the set $I' = I \cup [b+1, c-1]$ is independent with $|I'| > |I|$, which contradicts our choice of I.

Next we show that the gap between any two consecutive I-intervals is at most $k-2$. Suppose to the contrary that there exist consecutive I-intervals $[a,b]$ and $[c,d]$ such that $|[b+1, c-1]| \ge k-1$. Since $[a,b] \subseteq I$, it follows from the definition of $S_{k,k'}$ that $[a+k, b+k'] \cap I = \emptyset$. Hence $[b+1, b+k'] \cap I = \emptyset$. We partition the interval $[b+k'+1, b]$ $(= [0, n-1] - [b+1, b+k'])$ into subintervals of length $k+k'$, except the last subinterval which may have size less than $k+k'$ (when $r \ge k'+1$). If $0 \le r \le k'$, then the number of such subintervals is equal to q. By Lemma 2.6, $|I| \le qk$, which is contrary to our assumption. If $k'+1 \le r \le k+k'-1$, then the number of such subintervals is equal to $q+1$, and the last interval has size $r-k'$. Again, it follows from Lemma 2.6 that $|I| \le qk+r-k'$, which is contrary to our assumption. Therefore, the gap between any two consecutive I-intervals is at most $k-2$.

Now we show that the gap between any two consecutive I-intervals is greater than $2(k'-k)$. Assume to the contrary that $[a,b]$ and $[c,d]$ are consecutive I-intervals with gap t, $t \leq 2(k'-k)$. Let

$$I' = (I \cup [b+1, c-1]) - ([b+k'+1, c+k-1] \cup [b-k+1, c-k'-1]).$$

It is clear that I' is an independent set of G with $|I'| \geq |I| + t - 2(t-(k'-k)) \geq |I|$. Hence, I' is a maximum independent set with less intervals than I, which is contrary to our assumption.

We conclude that the gap between any two consecutive I-intervals is between $2(k'-k)+1$ and $k-2$. In particular, this implies that $2(k'-k)+1 \leq k-2$; otherwise, we have already arrived at a contradiction. Now, by the assumption that $k' \geq (5/4)k$, the gap between any two consecutive I-intervals is between $\frac{k}{2}+1$ and $k-2$. This implies that any set of k consecutive vertices in G intersects exactly two I-intervals.

For any consecutive I-intervals $[a,b]$ and $[c,d]$, we claim that $|[a,b]| + |[c,d]| \leq \frac{k}{2} - 1$. First we note that $|[a,d]| \leq k$. Otherwise, we would have $d \geq a+k$. This implies $c \geq b+k'+1$ since $[a+k, b+k'] \cap I = \emptyset$. Then the gap between $[a,b]$ and $[c,d]$ would be greater than $k-2$, which is a contradiction. It follows that $|[a,b]| + |[c,d]| \leq k - 2(k'-k) - 1 \leq \frac{k}{2} - 1$ since $k' \geq (5/4)k$.

Now choose two consecutive I-intervals $[a,b]$ and $[c,d]$ such that $|[a,b]| + |[c,d]|$ is the largest among all pairs of consecutive I-intervals. Let $[u,v]$ be the I-interval preceding $[a,b]$ (i.e., $[u,v]$ and $[a,b]$ are consecutive I-intervals) and let $[x,y]$ be the I-interval following $[c,d]$. Since $|I| > qk \geq k$ and since the union of any two consecutive I-intervals contains at most $\frac{k}{2}-1$ vertices, we know that there are at least five I-intervals. So the intervals $[a,b]$, $[c,d]$, $[u,v]$, and $[x,y]$ are distinct.

We now show that $[x,y]$ (respectively, $[u,v]$) is the only I-interval included in $[b+k'+1, c+k-1]$ (respectively, in $[b-k+1, c-k'-1]$). Because $[a,b], [c,d] \subseteq I$, we have $([a+k, b+k'] \cup [c+k, d+k']) \cap I = \emptyset$. In addition, by the arguments above, $[a,b]$, $[c,d]$ and $[c,d]$, $[x,y]$ are the only two I-intervals included in $[a, a+k-1]$ and $[c, c+k-1]$, respectively. Hence $[x,y] \subseteq [b+k'+1, c+k-1]$, and $[x,y]$ is the only I-interval included in $[b+k'+1, c+k-1]$. Similarly, we can show that $[u,v]$ is the only I-interval included in $[b-k+1, c-k'-1]$.

According to the choice of $[a,b]$ and $[c,d]$, we have $|[u,v]| \leq |[c,d]|$ and $|[x,y]| \leq |[a,b]|$. Therefore $|[u,v]| + |[x,y]| \leq |[a,b]| + |[c,d]| \leq \frac{k}{2} - 1$. Let

$$I' = (I \cup [b+1, c-1]) - ([u,v] \cup [x,y]).$$

By the discussion in the previous paragraph, it is clear that I' is an independent set with

$$|I'| \geq |I| + \frac{k}{2} + 1 - \left(\frac{k}{2} - 1\right) > |I|,$$

which contradicts our maximality assumption about I. □

Corollary 2.9. *If $k' \geq (5/4)k$, then $G(n, S_{k,k'})$ is star extremal.*

Theorem 2.10. *Suppose $G = G(n, S_{k,k'})$ with $n = q(k+k')+r$, $0 \leq r \leq k+k'-1$, and*

$$q > \frac{1}{k-k'} - \frac{k'}{k+k'} - \frac{kk'}{(k-k')(k+k')}.$$

Then G is star extremal. Moreover, the values of $\alpha(G)$ and $\lambda(G)$ are the same as in Theorem 2.8.

Proof. Let I be a maximum independent set of G. The I-intervals are similarly defined as in the proof of Theorem 2.8. Assume that the chosen set I has the minimum number of I-intervals among all maximum independent sets of G. Then the gap between any two consecutive I-intervals, as shown in the proof of Theorem 2.8, is between $2(k'-k)+1$ and $k-2$.

In the following, we show that for any i, $|I \cap [i, i+k+k'-1]| \leq 2k-k'$. Let a be the least element of $I \cap [i, i+k+k'-1]$ and let $[a,b]$ be the first nonempty intersection of an I-interval with $[i, i+k+k'-1]$. Note that when $a=i$, $[a,b]$ may be a part of an I-interval. Similarly, let $[c,d]$ be the last nonempty intersection of an I-interval with $[a, a+k]$. By the proof of Theorem 2.8, $[a,b]$ and $[c,d]$ are the two I-intervals included in $[a, a+k-1]$, so $[a,b] \neq [c,d]$.

In addition, it is true that $I \cap [c-(k'-k), c-1] = \emptyset$ since the gap between any two consecutive I-intervals is at least $2(k'-k) > k'-k$. Because $I \cap ([c+k, d+k'] \cup [a+k, a+k']) = \emptyset$, the following is clear:

$$I \cap [i, i+k+k'-1] \subset I \cap [a, a+k+k'-1] = [c,d] \cup (I \cap (A \cup B)),$$

where $A = [a, c-(k'-k)-1] \cup [d+1, a+k-1]$ and $B = [a+k', c+k-1] \cup [d+k'+1, a+k+k'-1]$. For each vertex $x \in A$, we have $x+k' \in B$. So if $x \in I \cap A$, then $x+k' \in B-I$, and vice versa. This one-to-one correspondence implies that $|I \cap (A \cup B)| \leq |A| = |B|$. Therefore, $|I \cap [i, i+k+k'-1]| \leq |A| + |[c,d]| = k-(k'-k) = 2k-k'$.

For each $0 \leq i \leq n-1$, let $n_i = |I \cap [i, i+k+k'-1]|$. Then

$$(k+k')|I| = \sum_{i=0}^{n-1} n_i \leq n(2k-k').$$

Now if $0 \leq r \leq k'$, then by Lemmas 2.1 and 2.7, it suffices to show that $\alpha(G) = |I| \leq qk$. Assume to the contrary that $|I| \geq qk+1$. Since $n \leq q(k+k')+k'$, we have

$$(k+k')(qk+1) \leq (q(k+k')+k')(2k-k').$$

A contradiction emerges after simplifying this inequality.

If $k' < r \leq k+k'-1$, it suffices to show that $|I| \leq qk+r-k'$. If $|I| \geq qk+r-k'+1$, then

$$(k+k')(qk+r-k'+1) \leq (q(k+k')+r)(2k-k').$$

This inequality leads to a contradiction, too. □

3. Circulant graphs and distance graphs. Circulant graphs and distance graphs are closely related. Given a finite set S of positive integers, the distance graph $G(Z,S)$ can be viewed as the limit of the sequence of circulant graphs $G(n,S)$ as n approaches infinity. Therefore, for a given S, if $G(n,S)$ is star extremal for all n, then $G(Z,S)$ is star extremal. However, the reverse of this implication is not always true. Take $S = \{1,3,4,5\}$. It is known [3] that $G(Z,S)$ is star extremal, while it was proved [10] that $\chi_f(10,S) = 5 < \chi_c(10,S) = 6$. Thus $G(10,S)$ is not star extremal.

In this section, we prove that, for a given S, if the distance graph $G(Z,S)$ is star extremal, then there exist infinitely many n such that the circulant graphs $G(n,S)$ are star extremal.

The fractional chromatic number $\chi_f(Z,S)$ for a distance graph $G(Z,S)$ has very close connections with T-coloring [13] and an earlier number theory problem about the density of sequences with missing differences. For references about these connections, we refer the reader to [4, 11, 14]. Among the results in [4], it is proved that $\chi_f(Z,S)$ always exists and is a rational number for any finite S.

A *homomorphism* (or edge-preserving map) from a graph G to another graph H is a mapping $f : V(G) \to V(H)$ such that $f(u)f(v) \in E(H)$ if $uv \in E(G)$. If such a homomorphism exists, we say that G admits a homomorphism to H and denote this by $G \to H$. If $G \to H$, then we have $\chi_f(G) \le \chi_f(H)$ and $\chi_c(G) \le \chi_c(H)$ by composition of functions.

LEMMA 3.1. *For a given S, $G(Z,S) \to G(n,S)$ for all $n \ge 2\max S$, where $\max S$ denotes the largest member of the set S.*

Proof. Define a mapping $f : Z \to [0, n-1]$ by $f(x) = x \bmod n$. It is easy to verify that f is a homomorphism from $G(Z,S)$ to $G(n,S)$. ∎

COROLLARY 3.2. *For a given S, $\chi_c(Z,S) \le \chi_c(n,S)$ and $\chi_f(Z,S) \le \chi_f(n,S)$ for all $n \ge 2\max S$.*

THEOREM 3.3. *If $G(Z,S)$ is star extremal for a given S, then there exists a positive integer m such that $G(km,S)$ is star extremal for any positive integer k.*

Proof. By a result in [4], we may assume $\chi_c(Z,S) = \chi_f(Z,S) = p/q$ for some rational number p/q. Let $d = \max S$. According to Corollary 3.2, it is enough to show that, for some $m \ge 2d$, there exists a (p,q)-coloring for any $G(km,S)$ because we would then have

$$p/q = \chi_f(Z,S) \le \chi_f(mk,S) \le \chi_c(mk,S) \le p/q.$$

Since $\chi_c(Z,S) = p/q$, by a result in [10], there exists a (p,q)-coloring $f : Z \to [0, p-1]$ of $G(Z,S)$. Partition nonnegative integers into blocks such that each block consists of p^d consecutive vertices. Consider the restriction of f to these blocks. By the pigeonhole principle, there exist two blocks with the same color sequence. Let x and y be the leading vertices of these two blocks such that $x < y$. Then $f(x+i) = f(y+i)$ for $0 \le i \le p^d - 1$. Let $m = y - x$. Define the mapping $f'(j) = f(x+j)$ for $0 \le j \le m-1$. It is clear that f' is a (p,q)-coloring for $G(m,S)$.

For $k \ge 2$, define a mapping $f'' : [0, km-1] \to [0, p-1]$ by $f''(v) = f'(v \bmod m)$. It is clear that f'' is a (p,q)-coloring for $G(km,S)$. ∎

4. Independence ratio. In this section, we discuss relations between the independence ratio and the fractional chromatic number of circulant graphs and distance graphs. Based on these relations, we give counterexamples to two conjectures of Collins [6].

Let $S = \{a_1, a_2, \ldots, a_l\}$ be a set of positive integers with $a_1 < a_2 < \cdots < a_l$. In her study of the asymptotic independence ratio of the circulant graphs $G(n,S)$, Collins introduced the *S-graph*, denoted by $G(S)$, which has vertex set $V = \{0, 1, 2, \ldots, a_1 + a_l - 1\}$ and edge set $E = \{uv : |u - v| \in S\}$. Note that $G(S)$ is not necessarily a circulant graph.

Given n and S, let $\mu(n,S) := \alpha(n,S)/n$ and $\mu(S) := \alpha(G(S))/(a_1 + a_l)$ denote the independence ratio of the circulant graph $G(n,S)$ and the S-graph $G(S)$, respectively. The *asymptotic independence ratio* $L(S)$ of a given set S is defined in [6] by

$$L(S) := \lim_{n\to\infty} \mu(n,S).$$

According to $(**)$, we have $\mu(n,S) = 1/\chi_f(n,S)$. Combining this with the fact that $\chi_f(Z,S) = \lim_{n\to\infty} \chi_f(n,S)$, the following result is obtained.

THEOREM 4.1. *$L(S) = 1/\chi_f(Z,S)$ for any given S.*

A set $S = \{a_1, a_2, \ldots, a_l\}$, $a_1 < a_2 < \cdots < a_l$, $l \geq 2$, is called *reversible* if $a_1 + a_l = a_2 + a_{l-1} = \cdots = a_{\lfloor \frac{l}{2} \rfloor} + a_{\lceil \frac{l}{2} \rceil}$. Collins [6] proved that $L(S) = \mu(S)$ if S is reversible, and she proposed the following conjecture.

CONJECTURE 1 (see [6]). *Suppose $S = \{a_1, a_2, \ldots, a_l\}$, $a_1 < a_2 < \cdots < a_l$, $l \geq 2$, is a reversible set. Then $\alpha(n, S) = \lfloor n\mu(S) \rfloor$ for any integer n satisfying $n > a_1 + 2a_l$.*

We now give a counterexample to Conjecture 1. The interval set $S_{k,k'}$ studied in section 2 is reversible. However, by Lemma 2.6 and Theorem 2.8, we have $\mu(S_{k,k'}) = k/(k + k')$ and $\alpha(n, S_{k,k'}) \neq \lfloor n\mu(S_{k,k'}) \rfloor$ when $k' \geq (5/4)k$, $n = q(k + k') + r$, and $r \geq k' + 1$.

For a nonreversible set S, Collins [6] gave two methods for constructing reversible sets from S. Let $S = \{a_1, a_2, \ldots, a_l\}$ and let $x = a_{l-1} + a_l$ and $y = a_1 + a_l$. Define $\hat{S} = S \cup (x - S)$ (here $x - S$ is the set $\{x - i \mid i \in S\}$) and $\tilde{S} = S \cup (y - S)$. Collins [6] showed that $L(S) \geq \max\{\mu(\hat{S}), \mu(\tilde{S})\}$ and proposed the following.

CONJECTURE 2 (see [6]). *$L(S) = \max\{\mu(\hat{S}), \mu(\tilde{S})\}$.*

For a counterexample to this conjecture, take $S = \{1, 2, 3, 6\}$. It is known [13] and easy to see that $\omega(Z, S) = \chi(Z, S) = 4$, so $\chi_f(Z, S) = \chi_c(Z, S) = 4$. Hence $L(S) = 1/4$. However, $\hat{S} = \{1, 2, 3, 6, 7, 8\}$, $\tilde{S} = \{1, 2, 3, 4, 5, 6\}$, $\mu(\hat{S}) = 2/9$, and $\mu(\tilde{S}) = 1/7$.

REFERENCES

[1] A. ÁDÁM, *Research Problem* 2-10, J. Combin. Theory, 2 (1967), p. 393.
[2] B. ALSPACH AND T. PARSONS, *Isomorphism of circulant graphs and digraphs*, Discrete Math., 25 (1979), pp. 97–108.
[3] G. J. CHANG, L. HUANG, AND X. ZHU, *Circular chromatic numbers and fractional chromatic numbers of distance graphs*, European J. Combin., 19 (1998), pp. 423–431.
[4] G. J. CHANG, D. D.-F. LIU, AND X. ZHU, *Distance graphs and T-coloring*, J. Combin. Theory Ser. B, 75 (1999), pp. 259–269.
[5] B. CODENOTTI, I. GERACE, AND S. VIGNA, *Hardness results and spectral techniques for combinatorial problems on circulant graphs*, IEEE Trans. Comput., 48 (1999), pp. 345–351.
[6] K. L. COLLINS, *Circulants and sequences*, SIAM J. Discrete Math., 11 (1998), pp. 330–339.
[7] H. S. M. COXETER, *Twelve Geometric Essays*, Southern Illinois Univ. Press, Carbondale/Edwardsville, IL, 1968.
[8] W. DEUBER AND X. ZHU, *The chromatic number of distance graphs*, Discrete Math., 165/166 (1997), pp. 195–204.
[9] G. GAO, Y. WANG, AND H. ZHOU, *Star chromatic numbers of some planar graphs*, J. Graph Theory, 27 (1998), pp. 33–42.
[10] G. GAO AND X. ZHU, *star extremal graphs and the lexicographic product*, Discrete Math., 152 (1996), pp. 147–156.
[11] J. R. GRIGGS AND D. D.-F. LIU, *The channel assignment problem for mutually adjacent sites*, J. Combin. Theory Ser. A, 68 (1994), pp. 169–183.
[12] S. KLAVŽAR, *On the fractional chromatic number and the lexicographic product of graphs*, Discrete Math., 185 (1998), pp. 259–263.
[13] D. D.-F. LIU, *T-colorings of graphs*, Discrete Math., 101 (1992), pp. 203–212.
[14] D. D.-F. LIU, *T-coloring and chromatic number of distance graphs*, Ars Combin., to appear.
[15] T. PARSONS, *Circulant graph imbeddings*, J. Combin. Theory Ser. B, 29 (1980), pp. 310–320.
[16] D. J. KLEITMAN, *Problem session*, in Applications of Discrete Mathematics, Proceedings of the Third Conference on Discrete Mathematics, Clemson University, Clemson, SC, May 14–16, 1986, R. D. Ringeisen and F. S. Roberts, eds., SIAM, Philadelphia, PA, 1988, pp. 221–222.
[17] E. R. SCHEINERMAN AND D. H. ULLMAN, *Fractional Graph Theory*, John Wiley, New York, 1997.
[18] A. F. SIDORENKO, *Triangle-free regular graphs*, Discrete Math., 91 (1991), pp. 215–217.
[19] H. YEH AND X. ZHU, *Coloring Some Circulant Graphs*, manuscript, 1999.
[20] A. VINCE, *Star chromatic number*, J. Graph Theory, 12 (1988), pp. 551–559.
[21] X. ZHU, *Star chromatic numbers and products of graphs*, J. Graph Theory, 16 (1992), pp. 557–569.
[22] X. ZHU, *Circular Chromatic Number, a Survey*, manuscript, 1997.

SIAM J. DISCRETE MATH.
Vol. 12, No. 4, pp. 500–523

GRAPH PARTITIONING AND CONTINUOUS QUADRATIC PROGRAMMING*

WILLIAM W. HAGER† AND YAROSLAV KRYLYUK†

Abstract. A continuous quadratic programming formulation is given for min-cut graph partitioning problems. In these problems, we partition the vertices of a graph into a collection of disjoint sets satisfying specified size constraints, while minimizing the sum of weights of edges connecting vertices in different sets. An optimal solution is related to an eigenvector (Fiedler vector) corresponding to the second smallest eigenvalue of the graph's Laplacian. Necessary and sufficient conditions characterizing local minima of the quadratic program are given. The effect of diagonal perturbations on the number of local minimizers is investigated using a test problem from the literature.

Key words. graph partitioning, min-cut, max-cut, quadratic programming, optimality conditions, graph Laplacian, edge separators, Fiedler vector

AMS subject classifications. 90C35, 90C27, 90C20

PII. S0895480199335829

1. Introduction.

This paper analyzes a continuous quadratic programming formulation for min-cut graph partitioning problems where we partition the vertices of a graph into disjoint sets satisfying specified size constraints, while minimizing the sum of the weights of edges connecting vertices in different sets. As a special case, the discrete quadratic programming formulation of Goemans and Williamson [22] for the max-cut problem is equivalent to a continuous quadratic program in which their discrete variables taking values -1 or $+1$ are replaced by continuous variables with values between -1 and $+1$. Graph partitioning problems arise in circuit board and microchip design, in other layout problems (see [33]), and in sparse matrix pivoting strategies. In parallel computing, graph partitioning problems arise when tasks are partitioned among processors in order to minimize the communication between processors and balance the processor load. For example, an application of graph partitioning to parallel molecular dynamics simulations is given in [44].

Another graph problem with a quadratic programming formulation is the maximum clique problem. In [36] Motzkin and Strauss show that the size of the largest clique in a graph can be obtained by solving a quadratic programming problem, while Gibbons et al. establish in [20] many interesting properties of this formulation.

A general approach for converting a discrete optimization problem to a continuous problem involves a diagonal perturbation. For example, subtracting a sufficiently large multiple of the identity from the quadratic cost matrix in the quadratic assignment problem yields a concave minimization problem whose local minimizers are extreme points of the feasible set, and whose global minimizers are solutions of the original discrete optimization problem (see the book [38, p. 26] by Pardalos and Rosen and the article [4] by Bazaraa and Sherali). One problem with this concave formulation of a discrete minimization problem is that the continuous problem can have many local minimizers. When a continuous optimization algorithm is applied, any of these local minima can trap the iterates. Our approach is related in the sense that we modify

*Received by the editors January 1, 1999; accepted for publication (in revised form) May 28, 1999; published electronically October 19, 1999. This research was supported by the National Science Foundation.

http://www.siam.org/journals/sidma/12-4/33582.html

†Department of Mathematics, University of Florida, Gainesville, FL 32611 (hager@math.ufl.edu).

the diagonal of the cost function. However, we are able to capture the solution of the discrete problem without modifying the cost function to the extent that it becomes concave. By restricting the size of the modification, the number of local minimizers that are candidates for a global minimizer is reduced substantially.

Various approaches to the graph partitioning problem appear in the literature. The seminal paper in this area is that of Kernighan and Lin [31] which presents the problem, application areas, and an exchange algorithm for obtaining approximate solutions. Four classes of algorithms have emerged for the graph partitioning problem:

(a) spectral methods, such as those in [29] and [40], where an eigenvector corresponding to the second smallest eigenvalue (Fiedler vector) of the graph's Laplacian is used to approximate the best partition;
(b) geometric methods, such as those in [21], [28], and [35], where geometric information for the graph is used to find a good partition;
(c) multilevel algorithms, such as those in [13], [14], [30], and [32], that first coarsen the graph, partition the smaller graph, then uncoarsen to obtain a partition for the original graph;
(d) optimization-based methods, such as those in [5], [6], [7], [18], and [45], where approximations to the best partitions are obtained by solving optimization problems.

See [3] for a survey of results in this area prior to 1995.

Here we focus on optimization-based formulations. Much of the earlier work in this area involves relaxations in which constraints are dropped in an optimization problem to obtain a tractable problem whose optimal solution is a lower bound for the optimal partition (see, for example, [6], [17], [41]). We also mention the work of Barnes [5] in which a spectral decomposition of the adjacency matrix is used with the solution of a related transportation problem (linear cost function and linear constraints) to approximate the best partition. In [7] a diagonal perturbation of the adjacency matrix is used to make it positive definite, and a Cholesky factorization of this perturbed matrix leads to a transportation problem whose solution again approximates the best partition. In contrast, our quadratic program is an exact formulation of the original problem in the sense that it has a minimizer corresponding to the best partition. Since the graph partitioning problem is NP-hard, this exact formulation is, in general, a difficult problem to solve.

In [18] Falkner, Rendl, and Wolkowicz present a quadratic optimization problem with both a quadratic constraint and linear equality and inequality constraints that is equivalent to the graph partitioning problem, and they solve (approximately) problems from the literature using the bundle-trust code of Schramm and Zowe. Their constraints are of the form

$$0 \leq x_i \leq 1, \quad \sum_{i=1}^{n} x_i = m, \quad \sum_{i=1}^{n} x_i^2 = m,$$

which force the solution vector to have 0/1 components. In [45] Wolkowicz and Zhao consider another variation of the quadratic constraint, requiring that $x_i^2 = x_i$, to enforce the 0/1 constraint. A semidefinite programming relaxation of the original problem is solved using a primal-dual interior point method. Our quadratic programming formulation does not have a quadratic constraint; the constraints are simply linear equalities and inequalities. We show that the quadratic program has a solution with 0/1 components, and that there is a connection between the Fiedler vector used by Pothen, Simon, and Liou in [40] to compute edge and vertex separators of small size and a solution to our quadratic programming problem. Our proof of the existence of a 0/1 solution is based on the following principle exposed by Tardella in [43]: If a function is minimized over a polyhedron, and if for each face of the polyhedron there exists a direction in the face along which the function is concave (or quasi concave), then there exists a vertex minimizer.

We briefly outline the paper: Section 2 presents the quadratic programming formulation of the two-set graph partitioning problem. In section 3 we give necessary and sufficient optimality conditions for a local minimizer of the quadratic program. These conditions relate the graph structure and the first-order optimality conditions at the given point. In section 4 we examine the effect of diagonal perturbations on the number of local minimizers using a test problem of Donath and Hoffman [17]. The connection between our quadratic program and the second eigenvector of the graph's Laplacian is studied in section 5. In section 6 we conclude with various generalizations of our results to partitions involving more than two sets, to nonsymmetric matrices, and to more general constraints.

2. Two-set partitions. Let G be a graph with n vertices V:

$$V = \{1, 2, \ldots, n\},$$

and let a_{ij} be a weight associated with the edge (i, j). For each i and j, we assume that $a_{ii} = 0$, $a_{ij} = a_{ji}$, and if there is no edge between i and j, then $a_{ij} = 0$. The sign of the weights is not restricted. Given a positive integer $m < n$, we wish to partition the vertices into two disjoint sets, one with m vertices and the other with $n - m$ vertices, while minimizing the sum of the weights associated with edges connecting vertices in different sets. This optimal partition is called a min-cut. We show that for an appropriate choice of the diagonal matrix $\mathbf{D}$, the min-cut can be obtained by solving the following quadratic programming problem:

$$\text{(1)} \qquad \begin{array}{rl} \text{minimize} & (\mathbf{1} - \mathbf{x})^{\mathsf{T}}(\mathbf{A} + \mathbf{D})\mathbf{x} \\ \text{subject to} & \mathbf{0} \le \mathbf{x} \le \mathbf{1}, \;\; \mathbf{1}^{\mathsf{T}}\mathbf{x} = m. \end{array}$$

More precisely, for an appropriate choice of $\mathbf{D}$, (1) has a solution $\mathbf{y}$ for which each component is either 0 or 1. The two sets V_1 and V_2 in an optimal partition are given by

$$\text{(2)} \qquad V_1 = \{i : y_i = 1\} \quad \text{and} \quad V_2 = \{i : y_i = 0\}.$$

The following theorem shows how to choose $\mathbf{D}$.

THEOREM 2.1. *If* $\mathbf{D}$ *is chosen so that*

$$\text{(3)} \qquad d_{ii} + d_{jj} \ge 2a_{ij}$$

for each i and j, then (1) *has a* 0/1 *solution* $\mathbf{y}$ *and the partition given by* (2) *is a min-cut. Moreover, if for each i and j,*

$$\text{(4)} \qquad d_{ii} + d_{jj} > 2a_{ij},$$

then every local minimizer of (1) *is a* 0/1 *vector.*

Proof. Given a solution $\mathbf{y}$ to (1), we now construct a piecewise linear path, taking us from $\mathbf{y}$ to a solution $\mathbf{z}$ of (1) whose components are either 0 or 1. Let $\mathcal{F}(\mathbf{y})$ be the inactive (or free) components of the vector $\mathbf{y}$:

$$\text{(5)} \qquad \mathcal{F}(\mathbf{y}) = \{i : 0 < y_i < 1\}.$$

Let f be the cost function of (1):

$$f(\mathbf{x}) = (\mathbf{1} - \mathbf{x})^{\mathsf{T}}(\mathbf{A} + \mathbf{D})\mathbf{x}. \tag{6}$$

Either $\mathcal{F}(\mathbf{y})$ is empty, and $\mathbf{z} = \mathbf{y}$, or $\mathcal{F}(\mathbf{y})$ has two or more elements since the constraint $\mathbf{1}^{\mathsf{T}}\mathbf{x} = m$ of (1), where m is an integer, cannot be satisfied when $\mathbf{x}$ has a single noninteger component. If $\mathcal{F}(\mathbf{y})$ has two or more elements, we show that there exists another minimizing point $\bar{\mathbf{y}}$ with $\mathcal{F}(\bar{\mathbf{y}})$ strictly contained in $\mathcal{F}(\mathbf{y})$, and $f(\mathbf{x}) = f(\mathbf{y})$ for all $\mathbf{x}$ on the line segment connecting $\mathbf{y}$ and $\bar{\mathbf{y}}$. Utilizing this property in an inductive fashion, we conclude that there exists a piecewise linear path taking us from any given minimizer $\mathbf{y}$ to another minimizer $\mathbf{z}$ with $\mathcal{F}(\mathbf{z}) = \emptyset$ (that is, all the components of $\mathbf{z}$ are either 0 or 1), and $f(\mathbf{x}) = f(\mathbf{y})$ for all $\mathbf{x}$ on this path.

If $\mathcal{F}(\mathbf{y})$ has two or more elements, then choose two elements i and $j \in \mathcal{F}(\mathbf{y})$, and let $\mathbf{v}$ be the vector all of whose entries are zero except that $v_i = 1$ and $v_j = -1$. For ϵ sufficiently small, $\mathbf{x} = \mathbf{y} + \epsilon\mathbf{v}$ is feasible in (1). Expanding f in a Taylor series around $\mathbf{x} = \mathbf{y}$, we have

$$f(\mathbf{y} + \epsilon\mathbf{v}) = f(\mathbf{y}) - \epsilon^2\mathbf{v}^{\mathsf{T}}(\mathbf{A} + \mathbf{D})\mathbf{v}. \tag{7}$$

The $O(\epsilon)$ term in this expansion disappears since $f(\mathbf{y} + \epsilon\mathbf{v})$ achieves a minimum at $\epsilon = 0$, and the first derivative with respect to ϵ vanishes at $\epsilon = 0$. In addition, from the inequality

$$f(\mathbf{y} + \epsilon\mathbf{v}) \geq f(\mathbf{y}) \quad \text{for all } \ \epsilon \ \text{ near } 0,$$

we conclude that the quadratic term in (7) is nonnegative, or equivalently,

$$\mathbf{v}^{\mathsf{T}}(\mathbf{A} + \mathbf{D})\mathbf{v} \ = \ d_{ii}v_i^2 + d_{jj}v_j^2 + 2a_{ij}v_iv_j \ = \ d_{ii} + d_{jj} - 2a_{ij} \ \leq \ 0. \tag{8}$$

Since $d_{ii}+d_{jj}-2a_{ij} \geq 0$ by (3), it follows that $d_{ii}+d_{jj}-2a_{ij} = 0$ and $f(\mathbf{y}+\epsilon\mathbf{v}) = f(\mathbf{y})$ for each choice of ϵ. Let $\bar{\epsilon}$ be the largest value of ϵ for which $\mathbf{x} = \mathbf{y} + \epsilon\mathbf{v}$ is feasible in (1). Defining $\bar{\mathbf{y}} = \mathbf{y} + \bar{\epsilon}\mathbf{v}$, $\mathcal{F}(\bar{\mathbf{y}})$ is strictly contained in $\mathcal{F}(\mathbf{y})$ and $\bar{\mathbf{y}}$ achieves the minimum in (1) since $f(\mathbf{y} + \epsilon\mathbf{v}) = f(\mathbf{y})$ for all ϵ. In summary, for any given solution $\mathbf{y}$ to (1), we can find another solution $\bar{\mathbf{y}}$ with $\mathcal{F}(\bar{\mathbf{y}})$ strictly contained in $\mathcal{F}(\mathbf{y})$ and $f(\mathbf{x}) = f(\mathbf{y})$ for all $\mathbf{x}$ on the line segment connecting $\mathbf{y}$ and $\bar{\mathbf{y}}$. This shows that there exists a 0/1 solution $\mathbf{y}$ of (1).

Now, if $\mathbf{y}$ is a 0/1 vector, then $f(\mathbf{y})$ is equal to the sum of the weights of the edges connecting the sets V_1 and V_2 in (2). Conversely, given a partition of V into disjoint sets V_1 and V_2 and defining $z_i = 1$ for each $i \in V_1$ and $z_i = 0$ for each $i \in V_2$, $f(\mathbf{z})$ is the sum of the weights of the edges connecting V_1 and V_2. Combining these two observations, we conclude that the partition associated with $\mathbf{y}$ is a min-cut.

Finally, suppose that (4) holds, $\mathbf{y}$ is a local minimizer for (1), and $\mathbf{y}$ is not a 0/1 vector. As noted above, $\mathcal{F}(\mathbf{y})$ has two or more elements, and the expansion (7) holds where the quadratic term satisfies (8), contradicting (4). We conclude that $\mathcal{F}(\mathbf{y})$ is empty and $\mathbf{y}$ is a 0/1 vector. ∎

Note that condition (3) is equivalent to requiring that $\mathbf{f}$ in (6) is concave in the direction $\mathbf{v}$, where $\mathbf{v}$ is the vector all of whose entries are zero except that $v_i = 1$ and $v_j = -1$. Hence, concavity is not assumed over the entire space $\mathbf{R}^n$, only along directions corresponding to the edges of the constraint polyhedron. The technique we use in the proof of Theorem 2.1 to convert a noninteger minimizer to an integer minimizer by moving in the direction of the vector $\mathbf{v}$ is also employed by Ageev and

Sviridenko in [1]. Although the first part of Theorem 2.1 asserts the existence of a 0/1 solution to (1), there are instances where (1) has solutions that are not 0/1 vectors. For example, if the off-diagonal elements of $\mathbf{A}$ are all equal to 1 and $\mathbf{D} = \mathbf{I}$, then any feasible point is optimal.

We now consider a slightly more general form of the graph partitioning problem where we still minimize the sum of weights of edges connecting the two sets. However, the size of a set is specified by upper and lower bounds rather than by a fixed number m. Our quadratic programming formulation of this min-cut problem is the following:

$$\begin{aligned} &\text{minimize} \quad (\mathbf{1}-\mathbf{x})^{\mathsf{T}}(\mathbf{A}+\mathbf{D})\mathbf{x} \\ &\text{subject to} \;\; \mathbf{0} \le \mathbf{x} \le \mathbf{1}, \;\; l \le \mathbf{1}^{\mathsf{T}}\mathbf{x} \le u, \end{aligned} \tag{9}$$

where l and u are given integers satisfying $0 \le l < u \le n$. The corresponding generalization of Theorem 2.1 involves an additional constraint on the diagonal elements of $\mathbf{D}$.

COROLLARY 2.2. *If* $\mathbf{D}$ *is chosen so that*

$$d_{ii} + d_{jj} \ge 2a_{ij} \quad \textit{and} \quad d_{ii} \ge 0 \tag{10}$$

for each i and j, then (9) *has a* 0/1 *solution* $\mathbf{y}$ *and the partition given by* (2) *is a min-cut. Moreover, if for each i and j,*

$$d_{ii} + d_{jj} > 2a_{ij} \quad \textit{and} \quad d_{ii} > 0, \tag{11}$$

then every local minimizer of (9) *is a* 0/1 *vector.*

Proof. Let $\mathbf{y}$ be a solution of (9) and define $m = \mathbf{1}^{\mathsf{T}}\mathbf{y}$. If m is not an integer, then $l < m < u$ since m lies between the integers l and u. For $i \in \mathcal{F}(\mathbf{y})$, the free set defined in (5), let $\mathbf{e}$ be the vector all of whose entries are zero except that $e_i = 1$. The function $f(\mathbf{y}+\epsilon\mathbf{e})$, where f is defined in (6), has a local minimum at $\epsilon = 0$ since $\mathbf{y}$ is the global minimizer of (9) and $\mathbf{y} + \epsilon\mathbf{e}$ is feasible for small perturbations in ϵ. Expanding in a Taylor series around $\epsilon = 0$ gives

$$f(\mathbf{y}+\epsilon\mathbf{e}) = f(\mathbf{y}) - d_{ii}\epsilon^2. \tag{12}$$

Since $d_{ii} \ge 0$ by (10), it follows that $d_{ii} = 0$ or else the local optimality of $\mathbf{y}$ in (9) is violated. Hence,

$$f(\mathbf{y}+\epsilon\mathbf{e}) = f(\mathbf{y}) \tag{13}$$

for each choice of ϵ.

For each $i \in \mathcal{F}(\mathbf{y})$, we increase y_i until either y_i reaches the upper bound 1 or $\mathbf{1}^{\mathsf{T}}\mathbf{y}$ reach the upper bound u. These adjustments in y_i do not change the value of f due to (13), and after these adjustments, $\mathbf{1}^{\mathsf{T}}\mathbf{y}$ must be an integer. Therefore, without loss of generality, we can assume that $m = \mathbf{1}^{\mathsf{T}}\mathbf{y}$ is an integer. Since $\mathbf{y}$ is a solution of (9) and the feasible set of (1) is contained in the feasible set of (9), we conclude that $\mathbf{y}$ is a solution of (1) as well as (9). By Theorem 2.1, (1) has a 0/1 solution which must be a solution of (9).

Now suppose that (11) holds. If $\mathbf{1}^{\mathsf{T}}\mathbf{y}$ is not an integer, then we must have $l < \mathbf{1}^{\mathsf{T}}\mathbf{y} < u$. By (11) $d_{ii} > 0$ for each i. If $i \in \mathcal{F}(\mathbf{y})$, then according to (12) the local optimality of $\mathbf{y}$ is violated. Hence, we conclude that $\mathbf{1}^{\mathsf{T}}\mathbf{y}$ is an integer that we denote by m, and $\mathbf{y}$ is a local minimizer for (1) as well as for (9). By Theorem 2.1, $\mathbf{y}$ is a 0/1 vector. ∎

Remark 2.1. The proofs of Theorem 2.1 and Corollary 2.2 involve quadratic expansions of the cost function. The linear terms in these expansions all vanish due to the optimality of $\mathbf{y}$. Hence, both of these results are valid if linear terms are added to the cost functions in (1) and (9) since linear terms do not effect the quadratic terms in the expansions.

Now let us consider various applications of Theorem 2.1. If $a_{ij} = 1$ for each edge of the graph G, then $\mathbf{A}$ is simply the graph's adjacency matrix. And if $\mathbf{y}$ is a 0/1 vector, then $f(\mathbf{y})$ is equal to the number of edges connecting the sets V_1 and V_2 in (2) for the partition associated with $\mathbf{y}$. Notice that when $\mathbf{A}$ is the adjacency matrix of the graph, conditions (3) and (10) are satisfied by taking $\mathbf{D} = \mathbf{I}$.

Let $\mathbf{W}$ be an $n \times n$ symmetric matrix whose elements are nonnegative with $w_{ii} = 0$ for each i and consider the choice $\mathbf{A} = -\mathbf{W}$. Since $\mathbf{A} \leq \mathbf{0}$, it follows that the conditions (3) and (10) are satisfied by taking $\mathbf{D} = \mathbf{0}$. Hence, for this choice of $\mathbf{A}$ and for $\mathbf{D} = \mathbf{0}$, the quadratic programs (1) or (9) have 0/1 solutions. Since minimizing f is equivalent to maximizing $-f$, the minimization problem (9) is equivalent to the following max-cut problem:

$$\begin{aligned} &\text{maximize} \quad (\mathbf{1} - \mathbf{x})^{\mathsf{T}} \mathbf{W} \mathbf{x} \\ &\text{subject to} \quad \mathbf{0} \leq \mathbf{x} \leq \mathbf{1}, \ \ l \leq \mathbf{1}^{\mathsf{T}} \mathbf{x} \leq u. \end{aligned} \tag{14}$$

If $l = 0$ and $u = n$, then there are no constraints on the size of the sets in the partition. In [22] the following discrete formulation is given for the weighted max-cut problem without constraints on the set size:

$$\begin{aligned} &\text{maximize} \quad \frac{1}{2} \sum_{i<j} w_{ij} (1 - z_i z_j) \\ &\text{subject to} \quad z_i \in \{-1, 1\}, \ \ 1 \leq i \leq n. \end{aligned} \tag{15}$$

The cost function of this discrete quadratic program is equal to $\frac{1}{4}(\mathbf{1}^{\mathsf{T}} \mathbf{W} \mathbf{1} - \mathbf{z}^{\mathsf{T}} \mathbf{W} \mathbf{z})$, and with the substitution $\mathbf{z} = 2\mathbf{x} - \mathbf{1}$, we obtain the equivalent problem

$$\begin{aligned} &\text{maximize} \quad (\mathbf{1} - \mathbf{x})^{\mathsf{T}} \mathbf{W} \mathbf{x} \\ &\text{subject to} \quad x_i \in \{0, 1\}, \ \ 1 \leq i \leq n. \end{aligned} \tag{16}$$

Taking $l = 0$ and $u = n$, Corollary 2.2 implies that (14) has the same maximum as (16). Moreover, there exists a 0/1 solution $\mathbf{y}$ of (14) for which the associated partition (2) maximizes the sum of the weights of the edges connecting V_1 and V_2. As a consequence, if the constraint $z_i \in \{-1, 1\}$ in (15) is changed to $-\mathbf{1} \leq \mathbf{z} \leq \mathbf{1}$, then the resulting continuous quadratic program has the same maximum value as the discrete program (15). This property for bound-constrained minimization was observed by Rosenberg [42] in the following context: If a polynomial is linear with respect to each of its variables, then its minimum over a box is attained at one of the vertices. Since $w_{ii} = 0$, the function $\mathbf{z}^{\mathsf{T}} \mathbf{W} \mathbf{z}$ is linear in each variable and Rosenberg's result can be applied.

Graph partitioning problems have application to ordering strategies for sparse matrix factorization. In the minimum degree algorithm, we permute two rows and the same two columns of a symmetric positive definite matrix $\mathbf{P}$ in order to obtain as many zeros as possible in the first column. The column and the row that are moved to the first row and column correspond to the positive component of a 0/1 solution of (9) associated with $l = u = 1$, where $a_{ij} = 1$ if $p_{ij} \neq 0$ and $a_{ij} = 0$ otherwise. Likewise, taking $l = u = n/2$, assuming n is even, we obtain a partitioning akin to nested dissection in which all those columns and rows associated with indices in V_1 are permuted to the front of the matrix. Viewed in this graph partitioning context, another ordering emerges. For example, we could take $l = 1$ and u a number slightly larger than 1 to obtain an ordering similar to minimum degree. Or we could take $l < n/2$ and $u > n/2$ to obtain an ordering similar to nested dissection that allows some freedom in the size of the sets in the partition.

3. Necessary and sufficient optimality conditions. In this section, we formulate necessary and sufficient optimality conditions for the quadratic programs of section 2. For a general quadratic program, deciding whether a given point is a local minimizer is NP-hard (see [37], [39]). On the other hand, for the quadratic program associated with the graph partitioning problem, we show in this section that local optimality can be decided quickly. Given any $\mathbf{x}$ that is feasible in (1), let us define the sets

$$\mathcal{U}(\mathbf{x}) = \{i : x_i = 1\} \quad \text{and} \quad \mathcal{L}(\mathbf{x}) = \{i : x_i = 0\}.$$

Given a scalar λ, we define the vector

$$\boldsymbol{\mu}(\mathbf{x}, \lambda) = (\mathbf{A} + \mathbf{D})\mathbf{1} - 2(\mathbf{A} + \mathbf{D})\mathbf{x} + \lambda\mathbf{1}.$$

We also introduce subsets $\mathcal{U}_0$ and $\mathcal{L}_0$ defined by

$$\mathcal{U}_0(\mathbf{x}, \lambda) = \{i \in \mathcal{U}(\mathbf{x}) : \mu_i(\mathbf{x}, \lambda) = 0\} \quad \text{and} \quad \mathcal{L}_0(\mathbf{x}, \lambda) = \{i \in \mathcal{L}(\mathbf{x}) : \mu_i(\mathbf{x}, \lambda) = 0\}.$$

The first-order optimality (Karush–Kuhn–Tucker) conditions associated with a local minimizer $\mathbf{x}$ of (9) can be written in the following way: For some scalar λ,

$$\mathbf{0} \le \mathbf{x} \le \mathbf{1}, \quad \mathbf{1}^{\mathsf{T}}\mathbf{x} = m, \quad \text{and} \quad \mathbf{x} \in \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}, \lambda)), \tag{17}$$

where $\mathcal{N}(\boldsymbol{\mu}) = \mathcal{N}_1(\boldsymbol{\mu}) \times \mathcal{N}_2(\boldsymbol{\mu}) \times \cdots \times \mathcal{N}_n(\boldsymbol{\mu})$ is a set-valued map, and

$$\mathcal{N}_i(\boldsymbol{\mu}) = \begin{cases} R & \text{if } \mu_i = 0, \\ \{1\} & \text{if } \mu_i < 0, \\ \{0\} & \text{if } \mu_i > 0. \end{cases}$$

Here R denotes the set of real numbers. The first two conditions in (17) are the constraints in (1), while the last condition is complementary slackness and stationarity of the Lagrangian.

THEOREM 3.1. *Suppose that* (3) *holds and m is a real number with $0 < m < n$. A necessary and sufficient condition for $\mathbf{y}$ to be a local minimizer in* (1) *is that all of the following hold:*

(P1) *For some λ, the first-order conditions are satisfied at $\mathbf{x} = \mathbf{y}$.*
(P2) *For each i and $j \in \mathcal{F}(\mathbf{y})$, where $\mathcal{F}$ is the free index set defined in* (5), *we have $d_{ii} + d_{jj} = 2a_{ij}$.*
(P3) *Consider the three sets $\mathcal{U}_0(\mathbf{y}, \lambda)$, $\mathcal{L}_0(\mathbf{y}, \lambda)$, and $\mathcal{F}(\mathbf{y})$. For each i and j in two different sets, we have $d_{ii} + d_{jj} = 2a_{ij}$.*

The motivation for (P2) and (P3) follows. Those indices in $\mathcal{U}_0(\mathbf{y}, \lambda)$, $\mathcal{L}_0(\mathbf{y}, \lambda)$, and $\mathcal{F}(\mathbf{y})$ correspond to those components of the multiplier $\boldsymbol{\mu}(\mathbf{y}, \lambda)$ that vanish. If the cost function $f(\mathbf{x})$ in (6) is expanded in a Taylor series around $\mathbf{y}$, then the linear terms in the expansion corresponding to zero multiplier components are all zero. If $\mathbf{v}$ is a vector all of whose components are zero except that $v_i = 1$ and $v_j = -1$, where i and j are indices corresponding to multiplier components that vanish, then $\mathbf{v}^{\mathsf{T}}(\mathbf{A}+\mathbf{D})\mathbf{v} \geq 0$ by (3). Conditions (P2) and (P3) are devised so that $\mathbf{v}^{\mathsf{T}}(\mathbf{A}+\mathbf{D})\mathbf{v} = 0$ whenever $\mathbf{v}$ is a feasible direction at $\mathbf{y}$ (if $\mathbf{v}^{\mathsf{T}}(\mathbf{A}+\mathbf{D})\mathbf{v} > 0$, then $\mathbf{y}$ is no longer a local minimizer).

Proof. If $\mathbf{y}$ is a local minimizer in (1), then the first-order conditions (17) hold automatically, while in the proof of Theorem 2.1, we saw that $d_{ii} + d_{jj} = 2a_{ij}$ for each i and $j \in \mathcal{F}(\mathbf{y})$ — see the discussion around (8). For the remainder of the proof, we let $\boldsymbol{\mu}$ and the various sets $\mathcal{L}$, $\mathcal{U}$, $\mathcal{F}$, $\mathcal{L}_0$, and $\mathcal{U}_0$ stand for $\boldsymbol{\mu}(\mathbf{y}, \lambda)$, $\mathcal{L}(\mathbf{y})$, $\mathcal{U}(\mathbf{y})$, $\mathcal{F}(\mathbf{y})$, $\mathcal{L}_0(\mathbf{y}, \lambda)$, and $\mathcal{U}_0(\mathbf{y}, \lambda)$, respectively. We also define complementary sets

$$\mathcal{L}' = \mathcal{L} \setminus \mathcal{L}_0 \quad \text{and} \quad \mathcal{U}' = \mathcal{U} \setminus \mathcal{U}_0.$$

$\mathcal{L}'$ is the set of indices for which $y_i = 0$ and $\mu_i > 0$, while $\mathcal{U}'$ is the set of indices for which $y_i = 1$ and $\mu_i < 0$.

To establish (P3), we expand the cost function in a Taylor series around $\mathbf{y}$. Let L be the Lagrangian defined by

$$L(\mathbf{x}) = f(\mathbf{x}) + \lambda(\mathbf{1}^{\mathsf{T}}\mathbf{x} - m) - \sum_{i \in \mathcal{L}} \mu_i x_i - \sum_{i \in \mathcal{U}} \mu_i (x_i - 1),$$

where f is the cost function in (6). By the complementary slackness condition in (17) and by the definition of $\boldsymbol{\mu}$, we have $L(\mathbf{y}) = f(\mathbf{y})$ and $\nabla L(\mathbf{y}) = 0$. Expanding the Lagrangian around $\mathbf{y}$, we have

$$L(\mathbf{y}+\mathbf{z}) = L(\mathbf{y}) + \nabla L(\mathbf{y})\mathbf{z} + \frac{1}{2}\mathbf{z}^{\mathsf{T}}\nabla^2 L(\mathbf{y})\mathbf{z} = f(\mathbf{y}) - \mathbf{z}^{\mathsf{T}}(\mathbf{A}+\mathbf{D})\mathbf{z}.$$

It follows that

$$\begin{aligned} f(\mathbf{y}+\mathbf{z}) &= L(\mathbf{y}+\mathbf{z}) - \lambda(\mathbf{1}^{\mathsf{T}}(\mathbf{y}+\mathbf{z}) - m) + \sum_{i \in \mathcal{L}} \mu_i (y_i + z_i) + \sum_{i \in \mathcal{U}} \mu_i (y_i + z_i - 1) \\ &= f(\mathbf{y}) - \mathbf{z}^{\mathsf{T}}(\mathbf{A}+\mathbf{D})\mathbf{z} - \lambda \mathbf{1}^{\mathsf{T}}\mathbf{z} + \sum_{i \in \mathcal{L}} \mu_i z_i + \sum_{i \in \mathcal{U}} \mu_i z_i. \end{aligned} \tag{18}$$

Suppose that $i \in \mathcal{U}_0$ and $j \in \mathcal{F}$ and let $\mathbf{v}$ be the vector all of whose entries are zero except that $v_i = -1$ and $v_j = 1$. The vector $\mathbf{x} = \mathbf{y} + \epsilon\mathbf{v}$ satisfies the constraints of (1) for ϵ sufficiently small, and by the definition of $\mathcal{U}_0$, $\mu_i = 0$. By (18), we have

$$f(\mathbf{y}+\epsilon\mathbf{v}) = f(\mathbf{y}) - \epsilon^2(d_{ii} + d_{jj} - 2a_{ij}).$$

Since $\mathbf{y}$ is a local optimizer in (1), we must have $d_{ii} + d_{jj} \leq 2a_{ij}$; while by (3), $d_{ii} + d_{jj} \geq 2a_{ij}$. Hence, $d_{ii} + d_{jj} = 2a_{ij}$. A similar argument can be used for all the other possible ways of choosing i and j from different sets $\mathcal{U}_0$, $\mathcal{L}_0$, and $\mathcal{F}$. This completes the proof of (P3).

Now consider the converse. That is, we assume that (P1)–(P3) all hold and we wish to show that $\mathbf{y}$ is a local minimizer in (1). Suppose that $\mathbf{x}$ satisfies the constraints of (1) and define $\mathbf{z} = \mathbf{x} - \mathbf{y}$, so that $\mathbf{x} = \mathbf{y} + \mathbf{z}$. Let $\mathcal{Z}$ denote the set defined by

$$\mathcal{Z} = \mathcal{F} \cup \mathcal{L}_0 \cup \mathcal{U}_0 = \{i : \mu_i = 0\}, \tag{19}$$

and let $\mathcal{Z}'$ be the complement:

$$\mathcal{Z}' = \mathcal{L}' \cup \mathcal{U}' = \{i : \mu_i \neq 0\}. \tag{20}$$

$$
\begin{array}{c} \\ \mathcal{F} \\ \mathcal{L}_0 \\ \mathcal{U}_0 \\ \mathcal{Z}' \end{array}
\begin{array}{c} \begin{array}{cccc} \mathcal{F} & \mathcal{L}_0 & \mathcal{U}_0 & \mathcal{Z}' \end{array} \\
\left[\begin{array}{cccc} = & = & = & ? \\ = & ? & = & ? \\ = & = & ? & ? \\ ? & ? & ? & ? \end{array} \right] \end{array}
$$

FIG. 3.1. *Structure of* $\mathbf{A} + \mathbf{D}$.

In the case that $\mathcal{Z}'$ is nonempty, we define the parameter

$$\sigma = \min \{|\mu_i| : i \in \mathcal{Z}'\},$$

which is positive by the definition of $\mathcal{Z}'$. For the remainder of the proof, we assume that $\mathcal{Z}'$ is nonempty, and at the end of the proof, we point out the adjustments that are needed to handle the case where $\mathcal{Z}'$ is empty. Since $\mathbf{x} = \mathbf{y} + \mathbf{z}$ satisfies the constraints in (1), we have $z_i \geq 0$ and $z_j \leq 0$ for all $i \in \mathcal{L}$ and $j \in \mathcal{U}$. Since $\mu_i \geq 0$ and $\mu_j \leq 0$ for all $i \in \mathcal{L}$ and $j \in \mathcal{U}$, it follows that

$$\sum_{i \in \mathcal{L}} \mu_i z_i + \sum_{j \in \mathcal{U}} \mu_j z_j = \sum_{i \in \mathcal{Z}'} \mu_i z_i \geq \sigma \sum_{i \in \mathcal{Z}'} |z_i|. \tag{21}$$

If $\|\cdot\|$ and $\|\cdot\|_{\mathcal{Z}'}$ denote the vector 1-norms defined by

$$\|z\| = \sum_{i=1}^{n} |z_i| \quad \text{and} \quad \|z\|_{\mathcal{Z}'} = \sum_{i \in \mathcal{Z}'} |z_i|,$$

then the relation (21) can be expressed

$$\sum_{i \in \mathcal{L}} \mu_i z_i + \sum_{j \in \mathcal{U}} \mu_j z_j \geq \sigma \|z\|_{\mathcal{Z}'}. \tag{22}$$

Now let us consider the quadratic term in (18). The structure of $\mathbf{A}$ is depicted in Figure 3.1. In this figure, an equal sign means that for the elements in that part of the matrix, we have $d_{ii} + d_{jj} = 2a_{ij}$, while a question mark means that we do not know anything about the elements in that region. The equal sign in the $(\mathcal{F}, \mathcal{F})$ position corresponds to (P2) while the remaining six equal signs correspond to (P3).

We now make a careful study of the quadratic term in (18) which can be expressed

$$-\mathbf{z}^{\mathsf{T}}(\mathbf{A} + \mathbf{D})\mathbf{z} = -\sum_{i,j \in \mathcal{Z}} a_{ij} z_i z_j - \sum_{(i,j) \notin \mathcal{Z} \times \mathcal{Z}} a_{ij} z_i z_j - \sum_{i=1}^{n} d_{ii} z_i^2.$$

For those (i, j) that lie in the part of the matrix in Figure 3.1 corresponding to the equal signs, the relation $a_{ij} = (d_{ii} + d_{jj})/2$ holds. With this substitution, a little algebra reveals that

$$\begin{aligned} -\mathbf{z}^{\mathsf{T}}(\mathbf{A} + \mathbf{D})\mathbf{z} = &-d \left(\sum_{i \in \mathcal{Z}} z_i \right) + \frac{1}{2} \sum_{i,j \in \mathcal{L}_0} (d_{ii} + d_{jj} - 2a_{ij}) z_i z_j \\ &+ \frac{1}{2} \sum_{i,j \in \mathcal{U}_0} (d_{ii} + d_{jj} - 2a_{ij}) z_i z_j - \sum_{(i,j) \notin \mathcal{Z} \times \mathcal{Z}} a_{ij} z_i z_j - \sum_{i \in \mathcal{Z}'} d_{ii} z_i^2, \end{aligned} \tag{23}$$

where d is defined by

$$d = \sum_{i \in \mathcal{Z}} d_{ii} z_i. \tag{24}$$

Since $\mathbf{x}$ is feasible in (9), we have $z_i \geq 0$ for all $i \in \mathcal{L}_0$ and $z_i \leq 0$ for all $i \in \mathcal{U}_0$. Since $d_{ii} + d_{jj} \geq 2a_{ij}$ by (3), we deduce that

$$\sum_{i,j \in \mathcal{L}_0} (d_{ii} + d_{jj} - 2a_{ij}) z_i z_j + \sum_{i,j \in \mathcal{U}_0} (d_{ii} + d_{jj} - 2a_{ij}) z_i z_j \geq 0. \tag{25}$$

Hence, we have

$$-\mathbf{z}^{\mathsf{T}}(\mathbf{A} + \mathbf{D})\mathbf{z} \geq -d \left(\sum_{i \in \mathcal{Z}} z_i \right) - \sum_{(i,j) \notin \mathcal{Z} \times \mathcal{Z}} a_{ij} z_i z_j - \sum_{i \in \mathcal{Z}'} d_{ii} z_i^2. \tag{26}$$

Combining the lower bounds (22) and (26), we conclude from (18) that

$$f(\mathbf{y} + \mathbf{z}) \geq f(\mathbf{y}) + \sigma \|z\|_{\mathcal{Z}'} - d \left(\sum_{i \in \mathcal{Z}} z_i \right) - \sum_{(i,j) \notin \mathcal{Z} \times \mathcal{Z}} a_{ij} z_i z_j - \sum_{i \in \mathcal{Z}'} d_{ii} z_i^2. \tag{27}$$

Since both $\mathbf{x} = \mathbf{y}$ and $\mathbf{x} = \mathbf{y} + \mathbf{z}$ satisfy the constraint $\mathbf{1}^{\mathsf{T}}\mathbf{x} = m$, it follows that $\mathbf{1}^{\mathsf{T}}\mathbf{z} = 0$, from which we obtain the relation

$$\sum_{i \in \mathcal{Z}} z_i = - \sum_{i \in \mathcal{Z}'} z_i.$$

Taking absolute values gives

$$\left| \sum_{i \in \mathcal{Z}} z_i \right| \leq \|z\|_{\mathcal{Z}'}.$$

Also, observe that

$$|z_i z_j| \leq \|z\| \|z\|_{\mathcal{Z}'} \quad \text{when} \quad (i,j) \notin \mathcal{Z} \times \mathcal{Z}$$

since either $i \in \mathcal{Z}'$ or $j \in \mathcal{Z}'$. Combining these observations with (27) yields

$$f(\mathbf{y} + \mathbf{z}) \geq f(\mathbf{y}) + \|\mathbf{z}\|_{\mathcal{Z}'} \left(\sigma - c\|\mathbf{z}\| \right), \tag{28}$$

where c is a constant that can be bounded in terms of the elements of $\mathbf{A}$ and $\mathbf{D}$. Hence, when $\|\mathbf{z}\|$ is sufficiently small, $f(\mathbf{y}+\mathbf{z}) \geq f(\mathbf{y})$, which implies that $\mathbf{y}$ is a local minimizer of f.

To conclude, we consider the case where $\mathcal{Z}'$ is empty. In this case, all the components of $\boldsymbol{\mu}$ vanish by (19). Hence, the last two terms in the Taylor expansion (18) vanish, while the $\mathbf{1}^{\mathsf{T}}\mathbf{z}$ term vanishes since both $\mathbf{x} = \mathbf{y}$ and $\mathbf{x} = \mathbf{y} + \mathbf{z}$ satisfy the constraint $\mathbf{1}^{\mathsf{T}}\mathbf{x} = m$. For the quadratic term in (18), the first term in the identity (23) vanishes since $\mathbf{1}^{\mathsf{T}}\mathbf{z} = 0$, the next two terms are nonnegative by (25), and the last two terms are not present since the complement of $\mathcal{Z}$ is empty. Combining these observations, $f(\mathbf{y} + \mathbf{z}) \geq f(\mathbf{y})$ whenever $\mathbf{x} = \mathbf{y} + \mathbf{z}$ is feasible in (1). Hence, $\mathbf{y}$ is a global minimizer for (1). This completes the proof. ∎

Remark 3.1. For quadratic programming problems and a point $\mathbf{y}$ satisfying the first-order conditions (17), a necessary and sufficient condition for $\mathbf{y}$ to be a local minimizer is the copositivity of the quadratic cost matrix over a certain cone (see [12] and [15]). In our context, this copositivity condition is equivalent to the inequality

$$\mathbf{v}^{\mathsf{T}}(\mathbf{A}+\mathbf{D})\mathbf{v} \le 0$$

whenever $\mathbf{v}$ lies in the set

$$\Gamma = \{\mathbf{v} \in \mathbf{R}^n : \mathbf{1}^{\mathsf{T}}\mathbf{v} = 0, v_i \le 0 \text{ if } y_i = 1, v_i \ge 0 \text{ if } y_i = 0, \mathbf{v}^{\mathsf{T}}(\mathbf{A}+\mathbf{D})(\mathbf{1}-2\mathbf{y}) = 0\}.$$

Utilizing the expansion (18), it can be shown that

$$\Gamma = \{\mathbf{v} \in \mathbf{R}^n : \mathbf{1}^{\mathsf{T}}\mathbf{v} = 0, v_i \le 0 \text{ if } i \in \mathcal{U}_0, v_i = 0 \text{ if } i \in \mathcal{U} \setminus \mathcal{U}_0 \text{ or } i \in \mathcal{L} \setminus \mathcal{L}_0, \\ v_i \ge 0 \text{ if } i \in \mathcal{L}_0\}.$$

With further analysis, analogous to that given in the proof of Theorem 3.1, the copositivity condition is equivalent to (P2) and (P3). Other references concerning copositivity and its application to optimality in quadratic programming include [8], [9], [10], [11], [16], [26], and [27].

Remark 3.2. Continuous optimization algorithms typically converge to a point $\mathbf{y}$ that satisfies the first-order conditions (17). Theorem 3.1 provides two conditions (P2) and (P3) that can be checked to determine whether $\mathbf{y}$ is a local minimizer. Moreover, if $\mathbf{y}$ is not a local minimizer, then careful study of the proof of Theorem 3.1 reveals a direction of descent for the quadratic cost function. In particular, suppose that $d_{ii} + d_{jj} > 2a_{ij}$ for indices i and j described in either (P2) or (P3). Let $\mathbf{v}$ be a vector whose entries are all zero except for entries i and j which are chosen so that $v_i = -v_j$ and $|v_i| = 1$. From (18) it follows that

$$f(\mathbf{y} + \epsilon\mathbf{v}) = f(\mathbf{y}) - (d_{ii} + d_{jj} - 2a_{ij})\epsilon^2 \tag{29}$$

since all the terms linear in $\mathbf{z} = \epsilon\mathbf{v}$ vanish. In any of the following cases, we take $v_i = -1$ and $v_j = 1$: (a) $i, j \in \mathcal{F}(\mathbf{y})$ or (b) $i \in \mathcal{U}_0(\mathbf{y}, \lambda)$ and $j \in \mathcal{F}(\mathbf{y})$ or (c) $i \in \mathcal{U}_0(\mathbf{y}, \lambda)$ and $j \in \mathcal{L}_0(\mathbf{y}, \lambda)$. In the case that $i \in \mathcal{L}_0(\mathbf{y}, \lambda)$ and $j \in \mathcal{F}(\mathbf{y})$, we take $v_i = 1$ and $v_j = -1$. Choosing $\mathbf{v}$ in this way, $\mathbf{x} = \mathbf{y} + \epsilon\mathbf{v}$ is feasible in (1) for $\epsilon > 0$ sufficiently small and by (29) the value of the cost function is strictly smaller.

We now examine the case when a local minimizer is strict. If $\mathcal{V} \subset V$ is a collection of vertices from the graph, let $\mathcal{V}_i$ denote the set of edges formed by i and the elements of $\mathcal{V}$:

$$\mathcal{V}_i = \{(i, j) : j \in \mathcal{V}\}.$$

Given a collection of edges $\mathcal{E}$, let $|\mathcal{E}|$ denote the sum of the weights of the edges:

$$|\mathcal{E}| = \sum_{(i,j)\in\mathcal{E}} a_{ij}.$$

COROLLARY 3.2. *A feasible point* $\mathbf{y}$ *for* (1) *is a strict local minimizer if and only if* $\mathcal{F}(\mathbf{y}) = \emptyset$ *and*

$$\min_{i\in\mathcal{L}(\mathbf{y})} |\mathcal{L}_i(\mathbf{y})| - |\mathcal{U}_i(\mathbf{y})| > \max_{j\in\mathcal{U}(\mathbf{y})} |\mathcal{L}_j(\mathbf{y})| - |\mathcal{U}_j(\mathbf{y})|. \tag{30}$$

Proof. Suppose that $\mathbf{y}$ is a strict local minimizer for (1). That is, $f(\mathbf{x}) > f(\mathbf{y})$ when $\mathbf{x}$ is near $\mathbf{y}$ and $\mathbf{x}$ is feasible in (1). If $\mathcal{F}(\mathbf{y})$ is nonempty, then as seen in the proof of Theorem 2.1, $\mathcal{F}(\mathbf{y})$ has at least two elements. By (P2) of Theorem 3.1, $d_{ii} + d_{jj} = 2a_{ij}$ for each i and $j \in \mathcal{F}(\mathbf{y})$. Letting $\mathbf{v}$ be a vector whose elements are all zero except that $v_i = 1$ and $v_j = -1$, the expansion (7) implies that

$$f(\mathbf{y} + \epsilon\mathbf{v}) = f(\mathbf{y}) \tag{31}$$

for all choices of ϵ. Since this violates the assumption that $\mathbf{y}$ is a strict local minimizer, we conclude that $\mathcal{F}(\mathbf{y})$ is empty. By the first-order conditions (17), we have

$$(\mathbf{A1} - 2\mathbf{Ay})_i + \lambda \geq 0 \geq (\mathbf{A1} - 2\mathbf{Ay})_j + \lambda \tag{32}$$

for all $i \in \mathcal{L}(\mathbf{y})$ and $j \in \mathcal{U}(\mathbf{y})$. Since $\mathcal{F}(\mathbf{y})$ is empty, $(\mathbf{A1})_i = |\mathcal{L}_i(\mathbf{y})| + |\mathcal{U}_i(\mathbf{y})|$ and $(\mathbf{Ay})_i = |\mathcal{U}_i(\mathbf{y})|$. Hence, we have

$$(\mathbf{A1} - 2\mathbf{Ay})_i = |\mathcal{L}_i(\mathbf{y})| - |\mathcal{U}_i(\mathbf{y})|, \tag{33}$$

and (32) yields

$$|\mathcal{L}_i(\mathbf{y})| - |\mathcal{U}_i(\mathbf{y})| \geq |\mathcal{L}_j(\mathbf{y})| - |\mathcal{U}_j(\mathbf{y})|$$

for each $i \in \mathcal{L}(\mathbf{y})$ and $j \in \mathcal{U}(\mathbf{y})$. If equality holds, for some $i \in \mathcal{L}(\mathbf{y})$ and $j \in \mathcal{U}(\mathbf{y})$, then equality must hold in (32) as well:

$$(\mathbf{A1} - 2\mathbf{Ay})_i + \lambda = 0 = (\mathbf{A1} - 2\mathbf{Ay})_j + \lambda.$$

This implies that $i \in \mathcal{L}_0(\mathbf{y})$ and $j \in \mathcal{U}_0(\mathbf{y})$. By (P3) of Theorem 3.1, $d_{ii} + d_{jj} = 2a_{ij}$. Choosing $\mathbf{v}$ as we did earlier, $\mathbf{x} = \mathbf{y} + \epsilon\mathbf{v}$ is feasible in (1) for $\epsilon > 0$ sufficiently small, and (31) holds, which violates strict local optimality.

Conversely, suppose that $\mathcal{F}(\mathbf{y}) = \emptyset$ and (30) holds. In this case, we can choose λ such that

$$|\mathcal{L}_i(\mathbf{y})| - |\mathcal{U}_i(\mathbf{y})| + \lambda > 0 > |\mathcal{L}_j(\mathbf{y})| - |\mathcal{U}_j(\mathbf{y})| + \lambda$$

for each $i \in \mathcal{L}(\mathbf{y})$ and $j \in \mathcal{U}(\mathbf{y})$. Utilizing (33) gives

$$(\mathbf{A1} - 2\mathbf{Ay})_i + \lambda > 0 > (\mathbf{A1} - 2\mathbf{Ay})_j + \lambda,$$

for each $i \in \mathcal{L}(\mathbf{y})$ and $j \in \mathcal{U}(\mathbf{y})$. For this choice of λ, the first-order conditions (17) hold and both $\mathcal{L}_0(\mathbf{y}, \lambda)$ and $\mathcal{U}_0(\mathbf{y}, \lambda)$ are empty. Hence, the set $\mathcal{Z}'$ in (20) is simply

$$\mathcal{Z}' = \{1, 2, \ldots, n\}.$$

In this case, the lower bound (28) implies that $\mathbf{y}$ is a strict local minimizer. ∎

We now consider the quadratic program (9) with inequality constraints. In this case, the first-order KKT conditions are the following: For some λ,

$$\mathbf{0} \leq \mathbf{x} \leq \mathbf{1}, \quad l \leq \mathbf{1}^{\mathsf{T}}\mathbf{x} \leq u, \quad \mathbf{1}^{\mathsf{T}}\mathbf{x} \in \mathcal{M}(\lambda), \quad \text{and} \quad \mathbf{x} \in \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}, \lambda)), \tag{34}$$

where $\mathcal{M}(\lambda)$ is the set-valued map defined by

$$\mathcal{M}(\lambda) = \begin{cases} R & \text{if } \lambda = 0, \\ \{l\} & \text{if } \lambda < 0, \\ \{u\} & \text{if } \lambda > 0. \end{cases}$$

COROLLARY 3.3. *Suppose that* (10) *holds. A necessary and sufficient condition for* $\mathbf{y}$ *to be a local minimizer in* (9) *is that* (P1)–(P3) *hold along with the following additional condition:*

(P4) *In the case that* $\lambda = 0$ *in the first-order condition* (34), $d_{ii} = 0$ *for each* $i \in \mathcal{F}(\mathbf{y}) \cup \mathcal{L}_0 \cup \mathcal{U}_0$ *if* $l < \mathbf{1}^\mathsf{T}\mathbf{y} < u$, $d_{ii} = 0$ *for each* $i \in \mathcal{F}(\mathbf{y}) \cup \mathcal{U}_0$ *if* $\mathbf{1}^\mathsf{T}\mathbf{y} = u$, *and* $d_{ii} = 0$ *for each* $i \in \mathcal{F}(\mathbf{y}) \cup \mathcal{L}_0$ *if* $\mathbf{1}^\mathsf{T}\mathbf{y} = l$.

Proof. We use the notation introduced in the proof of Theorem 3.1. If $\mathbf{y}$ is a local minimizer in (9), then the first-order conditions (34) hold automatically for some scalar λ. Since $\mathbf{y}$ is a local minimizer in (1) with $m = \mathbf{1}^\mathsf{T}\mathbf{y}$, it follows from Theorem 3.1 that (P2) and (P3) hold as well. If $\lambda = 0$ and $\mathbf{e}$ is a vector whose components are all zero except that $e_i = 1$ for some $i \in \mathcal{Z}$, then the expansion (18) yields

$$f(\mathbf{y} + \epsilon\mathbf{e}) = f(\mathbf{y}) - \epsilon^2 d_{ii}. \tag{35}$$

It follows that the local optimality of $\mathbf{y}$ is violated unless $d_{ii} = 0$ in all the cases cited in (P4).

Conversely, let us assume that (P1)–(P4) all hold. We wish to show that $\mathbf{y}$ is a local minimizer in (9). Given a feasible point $\mathbf{x}$ for (9), define $\mathbf{z} = \mathbf{x} - \mathbf{y}$. In Theorem 3.1, $\mathbf{1}^\mathsf{T}\mathbf{z} = 0$ and consequently, the λ term in (18) disappeared. Now this term needs to be included on the right side of (27) to obtain

$$\begin{aligned} f(\mathbf{y} + \mathbf{z}) \geq f(\mathbf{y}) + \sigma\|z\|_{\mathcal{Z}'} - \lambda\mathbf{1}^\mathsf{T}\mathbf{z} \\ -d\left(\sum_{i\in\mathcal{Z}} z_i\right) - \sum_{(i,j)\notin\mathcal{Z}\times\mathcal{Z}} a_{ij}z_iz_j - \sum_{i\in\mathcal{Z}'} d_{ii}z_i^2. \end{aligned} \tag{36}$$

In the proof of Theorem 3.1, we showed that the last two terms in (36) can be bounded by $c\|\mathbf{z}\|\|\mathbf{z}\|_{\mathcal{Z}'}$. Moreover, utilizing the identity

$$\sum_{i\in\mathcal{Z}} z_i = \mathbf{1}^\mathsf{T}\mathbf{z} - \sum_{i\in\mathcal{Z}'} z_i,$$

(36) yields

$$f(\mathbf{y} + \mathbf{z}) \geq f(\mathbf{y}) + (\sigma - c\|\mathbf{z}\|)\|\mathbf{z}\|_{\mathcal{Z}'} - (d + \lambda)\mathbf{1}^\mathsf{T}\mathbf{z}. \tag{37}$$

If $l < \mathbf{1}^\mathsf{T}\mathbf{y} < u$, then $\lambda = 0$ by (34) and $d = 0$ by (P4). It follows from (37) that $\mathbf{y}$ is a local minimizer. If $\mathbf{1}^\mathsf{T}\mathbf{y} = u$, then $\mathbf{1}^\mathsf{T}\mathbf{z} \leq 0$ when $\mathbf{x} = \mathbf{y} + \mathbf{z}$ is feasible in (9). If $\lambda = 0$, then by (P4), we have

$$d = \sum_{i\in\mathcal{L}_0} d_{ii}z_i \geq 0$$

since $z_i \geq 0$ for each $i \in \mathcal{L}$. Again, by (37) and the relation $\mathbf{1}^\mathsf{T}\mathbf{z} \leq 0$, $\mathbf{y}$ is a local minimizer. If $\lambda > 0$, then by choosing $\|\mathbf{z}\|$ small enough that $d + \lambda > 0$, we see from (37) that $\mathbf{y}$ is a local minimizer (since the last term in (37) is nonnegative). The case $\mathbf{1}^\mathsf{T}\mathbf{y} = l$ is treated in an analogous fashion. This completes the proof. ∎

4. An example. Theorem 2.1 and Corollary 2.2 require that the diagonal elements of $\mathbf{D}$ should be chosen large enough to satisfy (3) and (10), respectively. On the other hand, as we now observe, choosing $\mathbf{D}$ too large can lead to a miserable optimization problem. In the case that $\mathbf{D} = s\mathbf{I}$, the quadratic program (1) becomes

$$\begin{aligned} &\text{minimize } (\mathbf{1} - \mathbf{x})^\mathsf{T}(\mathbf{A} + s\mathbf{I})\mathbf{x} \\ &\text{subject to } \mathbf{0} \leq \mathbf{x} \leq \mathbf{1}, \ \ \mathbf{1}^\mathsf{T}\mathbf{x} = m. \end{aligned} \tag{38}$$

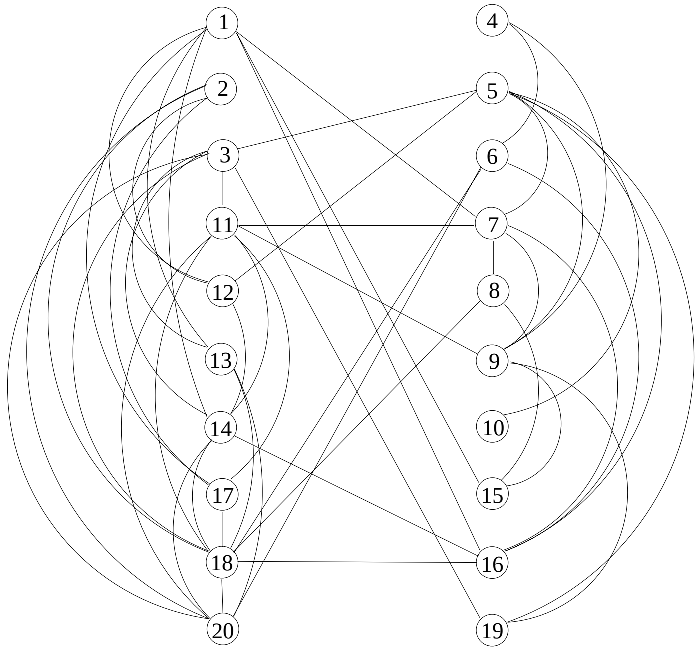


FIG. 4.1. *An example graph.*

Dividing the cost function in (38) by s and taking the limit as s tends to infinity, we obtain the problem

$$\begin{aligned} &\text{minimize } (\mathbf{1}-\mathbf{x})^{\mathsf{T}}\mathbf{x} \\ &\text{subject to } \ \mathbf{0} \le \mathbf{x} \le \mathbf{1}, \ \ \mathbf{1}^{\mathsf{T}}\mathbf{x} = m, \end{aligned} \tag{39}$$

The extreme points of the feasible set in either (1) or (38) or (39) is the set

$$\mathcal{X} = \{\mathbf{P1}_m : \mathbf{P} \in \mathcal{P}\},$$

where $\mathcal{P}$ is the set of $n \times n$ permutation matrices. Since $x = 0$ or $x = 1$ is a strict local minimizer of the function $x(1-x)$, we conclude that any element of $\mathcal{X}$ is a strict local minimizer in the problem (39). In fact, for s sufficiently large, any element of $\mathcal{X}$ is a strict local minimizer in the problem

$$\begin{aligned} &\text{minimize } (\mathbf{1}-\mathbf{x})^{\mathsf{T}}(\mathbf{I} + \tfrac{1}{s}\mathbf{A})\mathbf{x} \\ &\text{subject to } \ \mathbf{0} \le \mathbf{x} \le \mathbf{1}, \ \ \mathbf{1}^{\mathsf{T}}\mathbf{x} = m. \end{aligned}$$

Hence, as s tends to infinity in (38), every extreme point of the feasible set becomes a local minimizer, and consequently, checking the local minimizers in order to determine the global minimum involves checking every extreme point of the feasible set. As s decreases, fewer of these extreme points become local minimizers in (38), and there are fewer candidates for the global optimum.

As an illustration, let us consider the 20 node graph displayed in Figure 4.1 (see [17, Table 3], [41]) and let $\mathbf{A}$ be the adjacency matrix of the graph. In other words, the weight is 1 for each edge of the graph and 0 otherwise. For this choice of $\mathbf{A}$, we

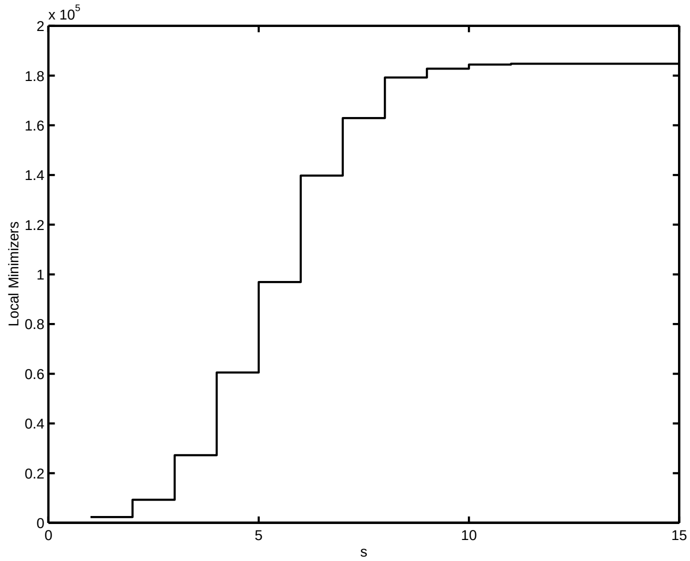


FIG. 4.2. *Number of local minimizers for Figure* 4.1 *graph and optimization problem* (38).

$(\mathbf{1}-\mathbf{x})^{\mathsf{T}}\mathbf{A}\mathbf{x}$	Minimizers
13	2
14	6
15	18
16	36
17	42
18	126
19	304

$(\mathbf{1}-\mathbf{x})^{\mathsf{T}}\mathbf{A}\mathbf{x}$	Minimizers
20	464
21	440
22	414
23	292
24	164
25	26

FIG. 4.3. *The number of local minimizers in* (1) *for each value of the cost function.*

should choose $s \geq 1$ in (38) to ensure that (3) holds. Following [17], we take $m = 10$, in which case the minimum number of edges separating the two sets of 10 nodes is 13 (the optimal partitioning is shown in Figure 4.1). For the example of Figure 4.1, we computed all the local minimizers of (38) for each value of $s \geq 1$. As s increases, the number of local minimizers increases monotonically. The values of s, where there is a change in the number, are always integers. Figure 4.2 shows the total number of local minimizers as a function of s. The number of local minimizers ranges from 2334 when $1 < s < 2$ up to 184756 for $s \geq 13$. Hence, there are about 79 times as many local minimizers for (38) when $s \geq 13$ as compared to the number of local minimizers when s is between 1 and 2.

For s between 1 and 2, the 2334 local minimizers of (38) yield the distribution of values for the cost function of (1) shown in Figure 4.3. Hence, out of the 2334 local minimizers of (1), only two of them are global minimizers. Note, however, that if we compute any local minimizer of (38), the largest value it can have is 25. Moreover, using 20 iterations of the gradient algorithm (optpack) described in [24], starting from a point near $\mathbf{x} = (m/n)\mathbf{1}$, we converge to a local minimizer of (1) with value $(\mathbf{1}-\mathbf{x})^{\mathsf{T}}\mathbf{A}\mathbf{x} = 14$. Hence, a simple gradient approach provides a partitioning of the vertices that is very close to the optimal partitioning $(\mathbf{1}-\mathbf{x})^{\mathsf{T}}\mathbf{A}\mathbf{x} = 13$.

In contrast, if we take $s = 21$, 11, 6, 3, and 2 in (38) and use exactly the same gradient algorithm and starting point, then we converge to locally minimizing values of 29, 26, 17, 15, and 14, respectively. Thus the smaller values of s yield computed minimizers whose values are closer to the global minimum 13.

The eigenvalues of the matrix $-(\mathbf{A}+\mathbf{I})$ are the following:

$$\begin{array}{rrrrrrr} -7.0429, & -4.1375, & -3.1908, & -2.7637, & -2.4979, & -2.2031, & -1.8808, \\ -1.7844, & -1.3706, & -1.0552, & -0.9066, & -0.4584, & 0.0901, & 0.2508, \\ 0.4315, & 1.0217, & 1.4759, & 1.8608, & 1.9740, & 2.1869 & \end{array}$$

Since there are both positive and negative eigenvalues, the choice $\mathbf{D} = \mathbf{I}$ in the quadratic program (1) has not changed the cost function to the extent that it became concave.

5. Graph eigenvectors. In [40], Pothen, Simon, and Liou propose using an eigenvector associated with the second largest eigenvalue of the Laplacian of a graph in order to compute edge and vertex separators of small size. In this section, we relate this eigenvector to a solution of the quadratic program (1). Let δ_i be the sum of the weights of edges emanating from vertex i:

$$\delta_i = \sum_{j=1}^{n} a_{ij}.$$

(Using the notation of section 3, $\delta_i = |V_i|$.) The Laplacian $\mathbf{L}$ associated with G is defined by

$$l_{ij} = \begin{cases} \delta_i & \text{if } i = j, \\ -a_{ij} & \text{otherwise.} \end{cases}$$

Let $g(\mathbf{x}) = \mathbf{x}^\mathsf{T}\mathbf{L}\mathbf{x}$ be the quadratic form associated with the Laplacian, and let f be the cost function of the quadratic program (1). See [45] for the first part of the following result.

PROPOSITION 5.1. *We have* $f(\mathbf{x}) = g(\mathbf{x})$ *for all* $\mathbf{x} \in \mathbf{\Omega}$, *where*

$$\mathbf{\Omega} = \{\mathbf{x} \in \mathbf{R}^n : x_i = 0 \text{ or } 1, \quad \mathbf{1}^\mathsf{T}\mathbf{x} = m\}.$$

Hence,

$$\min\{f(\mathbf{x}) : \mathbf{0} \le \mathbf{x} \le \mathbf{1},\ \mathbf{1}^\mathsf{T}\mathbf{x} = m\} \quad = \quad \min\{g(\mathbf{x}) : \mathbf{x} \in \mathbf{\Omega}\}.$$

Proof. Observe that $(\mathbf{1} - \mathbf{x})^\mathsf{T}\mathbf{A}\mathbf{x} = \boldsymbol{\delta}^\mathsf{T}\mathbf{x} - \mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x}$ and

$$\mathbf{x}^\mathsf{T}\mathbf{L}\mathbf{x} = \sum_{i=1}^{n} \delta_i x_i^2 - \mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x}.$$

It follows that

$$f(\mathbf{x}) - g(\mathbf{x}) = \sum_{i=1}^{n} \delta_i (x_i - x_i^2) = 0 \tag{40}$$

for every $\mathbf{x} \in \mathbf{\Omega}$. Since the quadratic program (1) has a solution in $\mathbf{\Omega}$ by Theorem 2.1, the proof is complete. □

By Gerschgorin's theorem (see [23, p. 341] or [25, p. 250]) $\mathbf{L}$ is positive semidefinite and clearly $\mathbf{1}$ is an eigenvector of $\mathbf{L}$ corresponding to the eigenvalue 0. Let $\mathbf{e}_i$, $i = 1, \ldots, n$, denote a linearly independent, normalized set of eigenvectors for $\mathbf{L}$, where $\mathbf{e}_1 = \mathbf{1}/\sqrt{n}$, and where the remaining eigenvectors are ordered so that for the associated eigenvalues, we have

$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n.$$

Since $g(\mathbf{1}) = 0$, it follows that for any vector $\mathbf{x}$, $g(\mathbf{x}) = g(\mathbf{Qx})$ where $\mathbf{Q}$ is the projection of a vector onto the orthogonal complement of $\mathbf{1}$. It is easily checked that

$$\mathbf{Qx} = \mathbf{x} - \tfrac{\mathbf{1}^{\mathsf{T}}\mathbf{x}}{n}\mathbf{1}.$$

Hence, if $\mathbf{x} \in \mathbf{\Omega}$, then

$$\mathbf{Qx} = \mathbf{x} - \tfrac{m}{n}\mathbf{1} \quad \text{and } \|\mathbf{Qx}\|_2 = \sqrt{\tfrac{m(n-m)}{n}}, \tag{41}$$

where $\|\cdot\|_2$ is the Euclidean norm. The function $g(\mathbf{y})$, with $\mathbf{y}$ restricted to a sphere in the orthogonal complement of $\mathbf{1}$, attains its minimum in the eigenspace associated with the second smallest eigenvalue λ_2. By the computation (41), all points of the form $\mathbf{Qx}$ with $\mathbf{x} \in \mathbf{\Omega}$ lie on the sphere of radius $R = \sqrt{m(n-m)/n}$. Hence, the problem of minimizing $g(\mathbf{x})$ over $\mathbf{x} \in \mathbf{\Omega}$ is related to the problem of finding the $\mathbf{x} \in \mathbf{\Omega}$ whose projection onto the orthogonal complement of $\mathbf{1}$ is closest to the eigenspace associated with λ_2.

In [40] the authors focus, in particular, on the case where the vertices are partitioned into two sets of roughly equal size. This case corresponds to taking $m = n/2$ in our notation. Since the eigenvectors associated with the second smallest eigenvalue are all orthogonal to $\mathbf{1}$, the average of the components for any of these eigenvectors is zero. If all the components are of comparable size, then half the components should be positive and the other half should be negative. The $\mathbf{x} \in \mathbf{\Omega}$ for which $\mathbf{Qx}$ is closest to a vector of this form is given by $x_i = 1$ for the positive components and $x_i = 0$ for the negative components. These considerations provide an alternative rationale for the methodology of [40] where the vertices are partitioned according to the sign of the components of an eigenvector corresponding to the second smallest eigenvalue.

This connection, provided by Proposition 5.1 between the quadratics f and g, leads to upper and lower bounds for f over $\mathbf{\Omega}$. In particular, since

$$\lambda_2 \|\mathbf{Qx}\|_2^2 \leq g(\mathbf{Qx}) \leq \lambda_n \|\mathbf{Qx}\|_2^2,$$

it follows from (41), Proposition 5.1, and the identity $g(\mathbf{x}) = g(\mathbf{Qx})$ for $\mathbf{x} \in \mathbf{\Omega}$ that

$$\lambda_2 R^2 \leq \min_{\mathbf{x} \in \mathbf{\Omega}} f(\mathbf{x}) \leq \max_{\mathbf{x} \in \mathbf{\Omega}} f(\mathbf{x}) \leq \lambda_n R^2, \tag{42}$$

where again $R = \sqrt{m(n-m)/n}$. The following lemma provides a small refinement to these bounds using adjacent eigenvalues:

LEMMA 5.2.

$$\lambda_3 R^2 - (\lambda_3 - \lambda_2) t_2 \leq \min_{\mathbf{x} \in \mathbf{\Omega}} f(\mathbf{x}) \leq \max_{\mathbf{x} \in \mathbf{\Omega}} f(\mathbf{x}) \leq \lambda_{n-1} R^2 + (\lambda_n - \lambda_{n-1}) t_n, \tag{43}$$

where

$$t_i = \max_{\mathbf{x} \in \mathbf{\Omega}} \ (\mathbf{e}_i^{\mathsf{T}} \mathbf{x})^2,$$

with $\mathbf{e}_i$ *the normalized eigenvector associated with the* i*th eigenvalue.*

Proof. We focus on the lower bound since exactly the same procedure can be applied to the upper bound. Given $\mathbf{x} \in \mathbf{\Omega}$, we let z_i denote the coordinates of $\mathbf{Qx}$ relative to the eigenvectors:

$$\mathbf{Qx} = \sum_{i=2}^{n} z_i \mathbf{e}_i.$$

By (41), we have

$$\sum_{i=2}^{n} z_i^2 = R^2 \quad \text{and} \quad z_2^2 = R^2 - \sum_{i=3}^{n} z_i^2.$$

By the definition of g, it follows that

$$\begin{aligned} g(\mathbf{Qx}) &= \sum_{i=2}^{n} \lambda_i z_i^2 \\ &= \lambda_2 R^2 + \sum_{i=3}^{n} (\lambda_i - \lambda_2) z_i^2 \\ &\geq \lambda_2 R^2 + (\lambda_3 - \lambda_2) \sum_{i=3}^{n} z_i^2 \\ &= \lambda_2 R^2 + (\lambda_3 - \lambda_2)(R^2 - z_2^2) \\ &= \lambda_3 R^2 - (\lambda_3 - \lambda_2) z_2^2. \end{aligned}$$

Since $z_2 = \mathbf{e}_2^{\mathsf{T}} \mathbf{x}$, it follows that $z_2^2 \leq t_2$. This completes the proof. ◻

In (42) and (43), we give bounds for the minimum and maximum of f over $\mathbf{\Omega}$ relative to the eigenvalues of the graph Laplacian $\mathbf{L}$. To the extent that the minimum or maximum in (42) or (43) can be evaluated, these inequalities can be used to obtain bounds on the eigenvalues themselves. For example, in the case $m = 1$, the minimum of f over $\mathbf{\Omega}$ is simply the minimum of δ_i, $1 \leq i \leq n$, while the maximum of f over $\mathbf{\Omega}$ is the largest of δ_i, $1 \leq i \leq n$. Letting $\underline{\delta}$ and $\bar{\delta}$ denote the minimum and maximum of the δ_i, we have the estimate (see [19])

$$\lambda_2 \leq \frac{n}{(n-1)} \underline{\delta} \leq \frac{n}{(n-1)} \bar{\delta} \leq \lambda_n.$$

6. Multiset generalizations. In the previous sections, we studied problems that were equivalent to partitioning the vertices of a graph G into two sets of given size, while minimizing the sum of the weights of edges connecting the sets. In this section, we consider the more general problem of partitioning the vertices into k distinct sets $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k$, with a given number of vertices $m_1, m_2, \ldots, m_k$ in each set, while minimizing the number of edges connecting different sets. Multiset partitions have application in VLSI design (see [3]) and in block iterative techniques for sparse linear systems, where rows and columns are permuted in order to minimize the number of nonzero elements outside the given diagonal blocks.

Let $\mathbf{X}$ be an $n \times k$ matrix, and let us define

$$x_{ij} = \begin{cases} 1 & \text{if } i \in S_j, \\ 0 & \text{if } i \notin S_j. \end{cases}$$

If $\mathbf{x}_j$ is the jth column of $\mathbf{X}$, then the expression $\mathbf{x}_j^\mathsf{T}\mathbf{A}\mathbf{x}_j$ equals the sum of the weights of edges connecting vertices in $\mathcal{S}_j$. The sum of the weights of edges connecting different sets in the partition is minimized when the sum of the weights of edges connecting vertices within the individual sets of the partition is maximized. Hence, the min-cut multiset partitioning problem is equivalent to the following discrete quadratic maximization problem:

$$\begin{array}{ll} \text{maximize} & \text{tr } \mathbf{X}^\mathsf{T}\mathbf{A}\mathbf{X} \\ \text{subject to} & \mathbf{X1} = \mathbf{1}, \quad \mathbf{X}^\mathsf{T}\mathbf{1} = \mathbf{m}, \quad \mathbf{X} \in \mathbf{\Lambda}, \end{array} \tag{44}$$

where tr denotes trace and

$$\mathbf{\Lambda} = \{\mathbf{X} \in \mathbf{R}^{nk} : x_{ij} = 0 \text{ or } 1, \quad 1 \le i \le n, \quad 1 \le j \le k\}.$$

The constraints $\mathbf{X1} = \mathbf{1}$ and $\mathbf{X} \in \mathbf{\Lambda}$ are equivalent to saying that each vertex is contained in precisely one of the sets $\mathcal{S}_j$. The constraint $\mathbf{X}^\mathsf{T}\mathbf{1} = \mathbf{m}$ is equivalent to saying that there are m_j vertices in set $\mathcal{S}_j$ for each j. This discrete quadratic programming formulation of the multiset partitioning problem can be found in [7], for example.

If $\mathbf{X}$ satisfies the constraints of (44), then

$$\text{tr } \mathbf{X}^\mathsf{T}\mathbf{D}\mathbf{X} = \sum_{j=1}^{k} \mathbf{x}_j^\mathsf{T}\mathbf{D}\mathbf{x}_j = \sum_{i=1}^{n} d_i.$$

Consequently, for any choice of the diagonal matrix $\mathbf{D}$, the problem (44) is equivalent to the following problem (since the cost functions differ by a constant, independent of the $\mathbf{X}$):

$$\begin{array}{ll} \text{maximize} & \text{tr } \mathbf{X}^\mathsf{T}(\mathbf{A}+\mathbf{D})\mathbf{X} \\ \text{subject to} & \mathbf{X1} = \mathbf{1}, \quad \mathbf{X}^\mathsf{T}\mathbf{1} = \mathbf{m}, \quad \mathbf{X} \in \mathbf{\Lambda}. \end{array} \tag{45}$$

Our goal in this section is to replace the discrete problem (44), where we impose the constraint $x_{ij} = 0$ or 1, by a continuous problem as in section 2. For example, in the special case $k = 2$, we seek to partition the vertices of the graph into two sets to maximize the total number of edges in the sets. The constraint $\mathbf{X1} = \mathbf{1}$ in (44) implies that $\mathbf{x}_2 = \mathbf{1} - \mathbf{x}_1$, and the cost function in (45) can be rewritten

$$\begin{aligned} \text{tr } \mathbf{X}^\mathsf{T}(\mathbf{A}+\mathbf{D})\mathbf{X} &= \mathbf{x}_1^\mathsf{T}(\mathbf{A}+\mathbf{D})\mathbf{x}_1 + \mathbf{x}_2^\mathsf{T}(\mathbf{A}+\mathbf{D})\mathbf{x}_2 \\ &= -2(\mathbf{1}-\mathbf{x}_1)^\mathsf{T}(\mathbf{A}+\mathbf{D})\mathbf{x}_1 + \mathbf{1}^\mathsf{T}(\mathbf{A}+\mathbf{D})\mathbf{1}. \end{aligned}$$

Hence, after negation and after identifying the $\mathbf{x}$ of (1) with $\mathbf{x}_1$, we see that the cost functions of (45) and of (1) differ only by a constant. Below, the notation $\mathbf{X} \ge \mathbf{0}$ means that every element of the matrix $\mathbf{X}$ is nonnegative.

THEOREM 6.1. *If* $\mathbf{D}$ *is chosen to satisfy* (3)*, then the continuous problem*

$$\begin{array}{ll} \textit{maximize} & \text{tr } \mathbf{X}^\mathsf{T}(\mathbf{A}+\mathbf{D})\mathbf{X} \\ \textit{subject to} & \mathbf{X1} = \mathbf{1}, \quad \mathbf{X}^\mathsf{T}\mathbf{1} = \mathbf{m}, \quad \mathbf{X} \ge \mathbf{0}, \end{array} \tag{46}$$

has a maximizer contained in $\mathbf{\Lambda}$*, and hence, this maximizer is a solution of the discrete problem*

$$\begin{array}{ll} \textit{maximize} & \text{tr } \mathbf{X}^\mathsf{T}(\mathbf{A}+\mathbf{D})\mathbf{X} \\ \textit{subject to} & \mathbf{X1} = \mathbf{1}, \quad \mathbf{X}^\mathsf{T}\mathbf{1} = \mathbf{m}, \quad \mathbf{X} \in \mathbf{\Lambda}. \end{array} \tag{47}$$

Conversely, every solution to (47) *is also a solution to* (46). *Moreover, if* (4) *holds, then every local maximizer for* (46) *lies in* $\mathbf{\Lambda}$.

Proof. Let $\mathbf{Y}$ denote any solution to (46), and let F denote the cost function defined by

$$F(\mathbf{X}) = \text{tr } \mathbf{X}^{\mathsf{T}}(\mathbf{A}+\mathbf{D})\mathbf{X}.$$

If an entry in $\mathbf{Y}$ lies on the open interval $I = (0,1)$, then we show that there exists another matrix $\bar{\mathbf{Y}}$ with the following properties:

(a) $\bar{\mathbf{Y}}$ is feasible in (46),

(b) $\bar{\mathbf{Y}}$ has at least one fewer entries contained in I than $\mathbf{Y}$, and

(c) $F(\mathbf{X}) = F(\mathbf{Y})$ for all $\mathbf{X}$ on the line segment connecting $\mathbf{Y}$ and $\bar{\mathbf{Y}}$.

Using these properties in an inductive fashion, we obtain, as in the proof of Theorem 2.1, a piecewise linear path taking us from $\mathbf{Y}$ to an optimal point $\mathbf{Z}$ for (46), and the elements of $\mathbf{Z}$ are either 0 or 1.

Proceeding with the construction, if $\mathbf{Y}$ has at least one entry in I, then by interchanging rows and columns if necessary, we can assume, without loss of generality, that $y_{11} \in I$. Since the column sums are integers, there is at least one more entry in column 1 of $\mathbf{Y}$ in I. (No entry of $\mathbf{Y}$ is larger than one since the row sums are all one.) Again, without loss of generality, we assume that $y_{21} \in I$. Since the row sums are integers, there is at least one more entry in the second row $\mathbf{Y}$ in I. Again, without loss of generality, we assume that $y_{22} \in I$.

Continuing this construction, we obtain the piecewise linear path depicted in Figure 6.1, where each point on the path corresponds to an index pair (i,j) for which $y_{ij} \in I$. Eventually, we reach an entry $y_{ij} \in I$ with the property that either the row index i or the column index j agrees with one of the predecessors. As depicted in Figure 6.1, we focus on the case where the row index i agrees with one of the predecessors; an analogous argument applies to the case where the column index agrees with that of a preceding column.

We discard the part of the path in Figure 6.1 that precedes the (i,i) element. Each entry of $\mathbf{Y}$ corresponding to an element of the path lies in I. Let $\mathbf{V}$ be the matrix that is entirely zero except for entries associated with elements on the path. We define $v_{ll} = 1$ for $i \le l \le j$, while the entries of $\mathbf{V}$ corresponding to the other elements on the path are all -1. Since the row and column sums of $\mathbf{V}$ all vanish, $\mathbf{Y} + \epsilon\mathbf{V}$ satisfies the linear constraints of (46) for any choice of ϵ. Since the elements of $\mathbf{Y}$ corresponding to points on the path in Figure 6.1 all lie in I, $\mathbf{Y} + \epsilon\mathbf{V} \ge \mathbf{0}$ for ϵ sufficiently close to 0.

Expanding in a Taylor series, we have

$$F(\mathbf{Y}+\epsilon\mathbf{V}) = F(\mathbf{Y}) + \epsilon^2 F(\mathbf{V}), \tag{48}$$

where the $O(\epsilon)$ term in the expansion vanishes since $F(\mathbf{Y}+\epsilon\mathbf{V})$ attains a local maximum at $\epsilon = 0$. By the structure of $\mathbf{V}$, we have

$$F(\mathbf{V}) = \left(d_{ii} + d_{jj} - 2a_{ij} + \sum_{l=i}^{j-1} d_{ll} + d_{l+1,l+1} - 2a_{l,l+1} \right). \tag{49}$$

By assumption (3), $F(\mathbf{V}) \ge 0$. If $F(\mathbf{V}) > 0$, then the optimality of $\mathbf{Y}$ is contradicted. Hence, $F(\mathbf{V}) = 0$, and we have $F(\mathbf{Y}+\epsilon\mathbf{V}) = F(\mathbf{Y})$ for all choices of ϵ. If $\bar{\epsilon}$ is the first value of ϵ for which a positive component of $\mathbf{Y}+\epsilon\mathbf{V}$ becomes zero, then the matrix

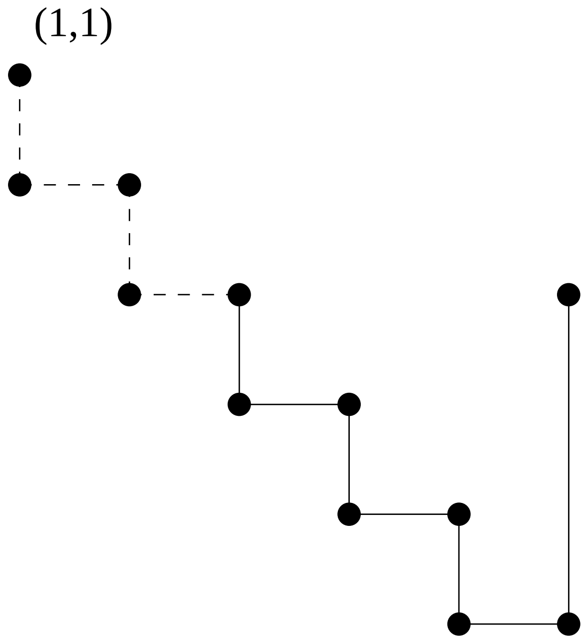


FIG. 6.1. *Indices of entries in* $\mathbf{Y}$ *that lie on the open interval* $(0, 1)$.

$\bar{\mathbf{Y}} = \mathbf{Y} + \bar{\epsilon}\mathbf{V}$ has at least one more zero than $\mathbf{Y}$. This completes the proof of (a)–(c) above.

Now suppose that $d_{ii} + d_{jj} > 2a_{ij}$ for all $i \neq j$ and let $\mathbf{Y}$ be any local maximizer. If $\mathbf{Y}$ has an element in I, then arguing as we did in the first part of the proof, we can construct a matrix $\mathbf{V}$ with elements 0, 1, and -1, whose nonzero elements correspond to elements of $\mathbf{Y}$ in I, and which satisfies (48) and (49). Since the right side of (49) is positive, we contradict the local optimality of $\mathbf{Y}$. Hence, each element of $\mathbf{Y}$ is either 0 or 1. ∎

Theorem 6.1 can be generalized in the following ways:

- Inspecting the proof, we utilize only the fact that the right side of the constraint $\mathbf{X1} = \mathbf{1}$ is an integer; the fact that the integer is 1 is used only to bound the components of $\mathbf{X}$ by 1. Hence, in (46) we can replace the right side of the constraint $\mathbf{X1} = \mathbf{1}$ with a more general vector of positive integers if we add the additional constraint $\mathbf{X} \leq \mathbf{1}$, where, in this matrix setting, $\mathbf{1}$ is the matrix whose elements are all 1.
- The proof of Theorem 6.1 also works if the cost function of (46) is replaced with

$$\sum_{l=1}^{k} \mathbf{x}_l^{\mathsf{T}} (\mathbf{A}_l + \mathbf{D}) \mathbf{x}_l,$$

  where $\mathbf{x}_l$ denotes column l of $\mathbf{X}$ and each $\mathbf{A}_l$ is symmetric matrix with zero diagonal that satisfies (3). (In circuit design, we may wish to associate different costs with the edges in different sets.)

- Since the proof of Theorem 6.1 utilizes a Taylor expansion of the quadratic cost function, a linear term can be added to the cost function without changing either the expansions or the conclusions.
- For a nonsymmetric matrix $\mathbf{A}$, we have

$$\text{tr } \mathbf{X}^\mathsf{T}\mathbf{A}\mathbf{X} = \tfrac{1}{2} \text{ tr } \mathbf{X}^\mathsf{T}\mathbf{A}\mathbf{X} + \tfrac{1}{2} \text{ tr } \mathbf{X}^\mathsf{T}\mathbf{A}^\mathsf{T}\mathbf{X} = \text{ tr } \mathbf{X}^\mathsf{T}\mathbf{S}\mathbf{X},$$

where $\mathbf{S} = \frac{1}{2}(\mathbf{A} + \mathbf{A}^\mathsf{T})$ is symmetric. Hence, Theorem 6.1 can be applied to the symmetric matrix $\mathbf{S}$ if the elements satisfy the condition (3). After making the substitution $s_{ij} = (a_{ij} + a_{ji})/2$, we see that the condition (3) of Theorem 6.1 is satisfied if

$$d_{ii} + d_{jj} \geq a_{ij} + a_{ji} \quad \text{for all } i \text{ and } j. \tag{50}$$

Collecting these observations, we have the following corollary.

COROLLARY 6.2. *If* $\mathbf{A}_l, l = 1, 2, \ldots, k$, *are* $n \times n$ *matrices, each of which satisfies the condition* (50), *and* $\mathbf{\Phi}$ *is a given* $k \times n$ *matrix, then the continuous problem*

$$\begin{aligned} &\textit{maximize} \quad \text{tr } \mathbf{\Phi}\mathbf{X} + \sum_{l=1}^{k} \mathbf{x}_l^\mathsf{T}(\mathbf{A}_l + \mathbf{D})\mathbf{x}_l \\ &\textit{subject to} \ \ \mathbf{X}\mathbf{1} = \mathbf{r}, \quad \mathbf{X}^\mathsf{T}\mathbf{1} = \mathbf{m}, \quad \mathbf{0} \leq \mathbf{X} \leq \mathbf{1}, \end{aligned} \tag{51}$$

where $\mathbf{r}$ *is a vector of positive integers, has a maximizer contained in* $\mathbf{\Lambda}$ *whenever the feasible set is nonempty, and hence, this maximizer is a solution of the discrete problem*

$$\begin{aligned} &\textit{maximize} \quad \text{tr } \mathbf{\Phi}\mathbf{X} + \sum_{l=1}^{k} \mathbf{x}_l^\mathsf{T}(\mathbf{A}_l + \mathbf{D})\mathbf{x}_l \\ &\textit{subject to} \ \ \mathbf{X}\mathbf{1} = \mathbf{r}, \quad \mathbf{X}^\mathsf{T}\mathbf{1} = \mathbf{m}, \quad \mathbf{X} \in \mathbf{\Lambda}. \end{aligned} \tag{52}$$

If $\mathbf{D}$ *also satisfies the strict inequality*

$$d_{ii} + d_{jj} > a_{ij}^l + a_{ji}^l \quad \textit{for all } i \neq j, \quad 1 \leq l \leq k,$$

where a_{ij}^l *is the* (i, j)*-element of* $\mathbf{A}_l$, *then every local maximizer of* (51) *lies in* $\mathbf{\Lambda}$.

Since the problem (52) with $\mathbf{r} = \mathbf{1}$ and $\mathbf{m} = \mathbf{1}$ is a special case of the quadratic assignment problem, Corollary 6.2 also provides an instance where the quadratic assignment problem can be replaced by a continuous quadratic programming problem whose Hessian is not necessarily positive definite. If $\mathbf{A}_l = \mathbf{0}$ for each l, then (51) is a linear programming problem with *transportation constraints* [34, p. 15]. If $\mathbf{A}_l = \mathbf{0}$ for each l, $\mathbf{X}$ is a square matrix, and $\mathbf{r} = \mathbf{m} = \mathbf{1}$, then (51) is the *linear assignment problem* [2, p. 215]. Hence, for these linear problems, Corollary 6.2 yields, as a special case, the existence of a 0/1 solution. For comparison, the existence of integer solutions in network flow problems can be found, for example, in [2, p. 245].

REFERENCES

[1] A. A. Ageev and M. I. Sviridenko, *Approximation algorithms for maximum coverage and max cut with given sizes of parts*, in Integer Programming and Combinatorial Optimization, Lecture Notes in Comput. Sci. 1610, G. C. Cornuéjols, R. E. Burkard, and G. J. Woeginger, eds., Springer, Berlin, 1999, pp. 17–30.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows*, in Optimization, Handbooks in Oper. Res. and Management Sci. 1, G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, eds., North–Holland, Amsterdam, 1989, pp. 211–369.

[3] C. J. Alpert and A. B. Kahng, *Recent directions in netlist partition: A survey*, Integration, the VLSI Journal, 19 (1995), pp. 1–81.

[4] M. S. Bazaraa and H. D. Sherali, *On the use of exact and heuristic cutting plane methods for the quadratic assignment problem,* J. Oper. Res. Soc., 33 (1982), pp. 991–1003.

[5] E. R. Barnes, *An algorithm for partitioning the nodes of a graph,* SIAM J. Alg. Discrete Methods, 3 (1982), pp. 541–550.

[6] E. R. Barnes and A. J. Hoffman, *Partitioning, spectra, and linear programming,* in Progress in Combinatorial Optimization, W. E. Pulleyblank, ed., Academic Press, New York, 1984, pp. 13–25.

[7] E. R. Barnes, A. Vannelli, and J. Q. Walker, *A new heuristic for partitioning the nodes of a graph,* SIAM J. Discrete Math., 1 (1988), pp. 299–305.

[8] I. M. Bomze, *Block pivoting and shortcut strategies for detecting copositivity*, Linear Algebra Appl., 248 (1996), pp. 161–184.

[9] I. M. Bomze, *Global escape strategies for maximizing quadratic forms over a simplex*, J. Global Optim., 11 (1997), pp. 325–338.

[10] I. M. Bomze and G. Danninger, *A global optimization algorithm for concave quadratic programming problems*, SIAM J. Optim., 3 (1993), pp. 826–842.

[11] I. M. Bomze and G. Danninger, *A finite algorithm for solving general quadratic problems*, J. Global Optim., 4 (1994), pp. 1–16.

[12] J. M. Borwein, *Necessary and sufficient conditions for quadratic minimality*, Numer. Funct. Anal. Optim., 5 (1982), pp. 127–140.

[13] T. N. Bui and C. Jones, *A heuristic for reducing fill-in in sparse matrix factorization,* in Proceedings of the 6th SIAM Conference on Parallel Processing for Scientific Computing, Vol. I, Norfolk, VA, March 22–24, 1993, R. F. Sincovec, D. E. Keyes, M. R. Leuze, L. R. Petzold, and D. A. Reed, eds., SIAM, Philadelphia, 1993, pp. 445–452.

[14] C.-K. Cheng and Y.-C. A. Wei, *An improved two-way partitioning algorithm with stable performance,* IEEE Trans. Comput. Aided Design, 10 (1991), pp. 1502–1511.

[15] L. B. Contesse, *Une caractérisation complète des minima locaux en programmation quadratique*, Numer. Math., 34 (1980), pp. 315–332.

[16] G. Danninger and I. M. Bomze, *Using copositivity for global optimality criteria in concave quadratic programming problems*, Math. Programming, 62 (1993), pp. 575–580.

[17] W. E. Donath and A. J. Hoffman, *Lower bounds for the partitioning of graphs,* IBM J. Res. Develop., 17 (1973), pp. 420–425.

[18] J. Falkner, F. Rendl, and H. Wolkowicz, *A computational study of graph partitioning,* Math. Programming, 66 (1994), pp. 211–240.

[19] M. Fiedler, *Algebraic connectivity of graphs,* Czechoslovak Math. J., 23 (1973), pp. 298–305.

[20] 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.

[21] J. R. Gilbert, G. L. Miller, and S.-H. Teng, *Geometric mesh partitioning: Implementation and experiments,* SIAM J. Sci. Comput., 19 (1998), pp. 2091–2110.

[22] 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.

[23] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1989.

[24] W. W. Hager, *Analysis and implementation of a dual algorithm for constrained optimization*, J. Optim Theory Appl., 79 (1993), pp. 427–462.

[25] W. W. Hager, *Applied Numerical Linear Algebra*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[26] J. B. Hiriart-Urruty, *Refinements of necessary optimality conditions in nondifferentiable programming.* I, Appl. Math. Optim., 5 (1979), pp. 63–82.

[27] J. B. Hiriart-Urruty, *Refinements of necessary optimality conditions in nondifferentiable programming.* II. *Optimality and stability in mathematical programming,* Math. Programming Stud., 19 (1982), pp. 120–139.

[28] M. T. Heath and P. Raghavan, *A Cartesian parallel nested dissection algorithm,* SIAM J. Matrix Anal. Appl., 16 (1995), pp. 235–253.

[29] B. Hendrickson and R. Leland, *An improved spectral graph partitioning algorithm for mapping parallel computations,* SIAM J. Sci. Comput., 16 (1995), pp. 452–469.

[30] B. Hendrickson and R. Leland, *A Multilevel Algorithm for Partitioning Graphs,* Technical Report SAND93-1301, Sandia National Laboratories, Albuquerque, NM, 1993.

[31] B. W. Kernighan and S. Lin, *An efficient heuristic procedure for partitioning graphs,* Bell System Tech. J., 49 (1970), pp. 291–307.

[32] G. Karypis and V. Kumar, *A fast and high quality multilevel scheme for partitioning irregular graphs,* SIAM J. Sci. Comput., 20 (1998), pp. 359–392.

[33] T. Lengauer, *Combinatorial Algorithms for Integrated Circuit Layout*, John Wiley, Chichester, 1990.

[34] D. G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, 1984.

[35] G. L. Miller, S.-H. Teng, W. Thurston, and S. A. Vavasis, *Automatic mesh partitioning,* in Graph Theory and Sparse Matrix Computations, J. Gilbert and J. Liu, eds., IMA Vol. Math. Appl. 56, Springer-Verlag, New York, 1993, pp. 57–84.

[36] T. S. Motzkin and E. G. Strauss, *Maxima for graphs and a new proof of a theorem of Turan,* Canad. J. Math., 17 (1965), pp. 533–540.

[37] K. G. Murty and S. N. Kabadi, *Some NP-complete problems in quadratic and linear programming*, Math. Programming, 39 (1987), pp. 117–129.

[38] P. M. Pardalos and J. B. Rosen, *Constrained Global Optimization: Algorithms and Applications,* Springer-Verlag, Berlin, 1987.

[39] P. M. Pardalos and S. A. Vavasis, *Quadratic programming with one negative eigenvalue is NP-hard*, J. Global Optim., 1 (1991), pp. 15–22.

[40] A. Pothen, H. D. Simon, and K.-P. Liou, *Partitioning sparse matrices with eigenvectors of graphs,* SIAM J. Matrix Anal. Appl., 11 (1990), pp. 430–452.

[41] F. Rendl and H. Wolkowicz, *A projection technique for partitioning the nodes of a graph*, Ann. Oper. Res., 58 (1995), pp. 172–191.

[42] I. Rosenberg, $0-1$ *optimization and nonlinear programming*, Rev. Française Automat. Informat. Recherche Opérationelle, 6 (1972), pp. 95–97.

[43] F. Tardella, *On the equivalence between some discrete and continuous optimization problems*, Ann. Oper. Res., 25 (1990), pp. 291–300.

[44] S.-H. Teng, *Provably good partitioning and load balancing algorithms for parallel adaptive N-body simulation*, SIAM J. Sci. Comput., 19 (1998), pp. 635–656.

[45] H. Wolkowicz and Q. Zhao, *Semidefinite programming relaxations for the graph partitioning problem,* Discrete Appl. Math., to appear.

SIAM J. DISCRETE MATH.
Vol. 12, No. 4, pp. 524-534

ON THE SIZE OF MINIMUM SUPER ARROVIAN DOMAINS*

SAMIT DASGUPTA†

Abstract. Arrow's celebrated impossibility theorem states that a sufficiently diverse domain of voter preference profiles cannot be mapped into social orders of the alternatives without violating at least one of three appealing conditions. Following Fishburn and Kelly, we define a set of strict preference profiles to be super Arrovian if Arrow's impossibility theorem holds for this set and each of its strict preference profile supersets. We write $\sigma(m, n)$ for the size of the smallest super Arrovian set for m alternatives and n voters. We show that $\sigma(m, 2) = \lceil \frac{2m}{m-2} \rceil$ and $\sigma(3, 3) = 19$. We also show that $\sigma(m, n)$ is bounded by a constant for fixed n and bounded on both sides by a constant times 2^n for fixed m. In particular, we find that $\lim_{n\to\infty} \sigma(3, n)/2^n = 3$. Finally, we answer two questions posed by Fishburn and Kelly on the structure of minimum and minimal super Arrovian sets.

Key words. Arrow's impossibility theorem, voter preference profiles, minimum profile sets

AMS subject classifications. 05A05, 90A08

PII. S0895480198332521

1. Introduction.

Arrow's impossibility theorem [1] states that a sufficiently diverse domain of voter preference profiles cannot be mapped into social orders of the alternatives without violating at least one of three appealing conditions. Fishburn and Kelly [2] consider the smallest domains of profiles of strict rankings for voters that induce an Arrovian dictator and have the property that every superset domain also induces an Arrovian dictator. In this paper, we continue their analysis and answer some of the questions left open in their work.

We consider a finite set X of $m \geq 3$ alternatives and a set of $n \geq 2$ voters, labeled $i = 1, \ldots, n$. Let $\mathbf{R}$ denote the set of all weak orders (transitive and complete binary relations that need not be asymmetric—that is, ties are allowed) on X, and let $\mathbf{S}$ be the set of all linear orders (strict rankings) on X. A *profile* of voter preferences is an n-tuple of strict rankings $d = (S_1, \ldots, S_n) \in \mathbf{S}^n$. Here the linear order S_i represents the preferences of voter i in the profile. A *domain* is a set of profiles: $D \subseteq \mathbf{S}^n$. A *social choice rule* on D is a mapping $f : D \to \mathbf{R}$ that assigns a weak order $\succsim_d$ on X to every $d \in D$. For $S \in \mathbf{S}$, the notation xSy means that x is preferred to y in S, and $S = x_1 x_2 \ldots x_m$ means that x_j is preferred to x_k whenever $j < k$. The strict part of a weak order $\succsim$ in $\mathbf{R}$ is denoted by $\succ$, that is, $x \succ y$ if $x \succsim y$ and $y \not\succsim x$. Given a subset of the alternatives $Y \subseteq X$, any weak order $\succsim$ in $\mathbf{R}$ induces a weak order $\succsim|_Y$ on Y. Similarly, any $S \in \mathbf{S}$ induces a linear order $S|_Y$ on Y. Given a profile $d = (S_1, \ldots, S_n)$, we write $d|_Y = (S_1|_Y, \ldots S_n|_Y)$ and say that d restricts to $d|_Y$.

A domain D is called *Arrovian* if there is no social choice rule f satisfying the following three conditions of Arrow [1], for all $x, y \in X$ and $d, e \in D$:

(P) *Pareto condition.* If $d = (S_1, \ldots, S_n)$ and xS_iy for $i = 1, \ldots, n$, then $x \succ_d y$.

(IIA) *Independence of irrelevant alternatives.* For $Y \subseteq X$, if $d|_Y = e|_Y$ then $\succsim_d|_Y$ is equal to $\succsim_e|_Y$.

*Received by the editors January 12, 1998; accepted for publication January 8, 1999; published electronically October 19, 1999. This research was supported by the National Science Foundation (grant DMS 9531373-001) and the National Security Agency (grant MDA-904-96-1-0044).
http://www.siam.org/journals/sidma/12-4/33252.html

†Department of Mathematics, Harvard University, Cambridge, MA 02138. Current address: 1916 Autumn Ridge Circle, Silver Spring, MD 20906 (dasgupta@post.harvard.edu).

(ND) *Nondictatorship.* There is no $i \in \{1,\ldots,n\}$ such that $\succ_d$ equals S_i for all $d = (S_1,\ldots,S_n) \in D$.

If $d = (S_1,\ldots,S_n)$ and the hypothesis of the Pareto condition xS_iy for $i = 1,\ldots,n$ holds, then we write $x \gg y$ in d. We say that x and y are a *Pareto pair* and are involved in a *Pareto relationship.* A mapping f that satisfies the first two conditions is called a (P) + (IIA) function. If the function f is such that $\succ_d$ equals S_i for each $d = (S_1,\ldots,S_n)$, then voter i is called a *dictator* for f. An Arrovian domain is one for which every (P) + (IIA) function has a dictator. A domain D is called *super Arrovian* if it is Arrovian and every superset domain $D' \supset D$ is also Arrovian. A super Arrovian domain on m alternatives and n voters that has the smallest number of profiles is called *minimum*, and this number of profiles is denoted $\sigma(m,n)$. A super Arrovian domain is called *minimal* if no proper subset is super Arrovian.

Fishburn and Kelly [2] proved that $\sigma(3,2) = 6$ and showed $\sigma(m,n) = o(4^n)$ for fixed m and $\sigma(m,n) = o((\log m)^{2+\epsilon})$ for fixed n and $\epsilon > 0$. They also proved $\sigma(m,2) \le \min\{6\cdot 2^{m-3}, (7\log m)^2\}$ and $\sigma(m,n) > 2^n - 2$. In this paper, we find improved lower and upper bounds and calculate the value of $\sigma(m,2)$ exactly.

THEOREM 1. *For $m \ge 3$, we have*

$$\sigma(m,2) = \left\lceil \frac{2m}{m-2} \right\rceil = \begin{cases} 6 & \text{if } m = 3, \\ 4 & \text{if } m = 4 \text{ or } 5, \\ 3 & \text{if } m \ge 6. \end{cases}$$

THEOREM 2. *For $m,n \ge 3$, we have*

$$\frac{m}{m-2}(2^n-2) < \sigma(m,n) \le \left\lceil \frac{2m}{m-2} \right\rceil (2^{n-1}-1) + \left\lceil \frac{2^n-2n-2}{\lfloor m/2 \rfloor} \right\rceil + \left\lceil \frac{n(n-1)}{2(m-2)} \right\rceil + 1.$$

For $n \ge 4$, the upper bound can be decreased by 1.

In Section 2, we establish the lower bound for $\sigma(m,n)$ given in Theorem 2. In Section 3, we prove an upper bound for $\sigma(m,n)$ in terms of $\sigma(m,2)$. In Section 4, we find a general class of super Arrovian domains which yields a tighter upper bound in the case $m = 3$. In Section 5 we evaluate $\sigma(m,2)$ explicitly, completing the proofs of Theorems 1 and 2. In Section 6, we show that $\sigma(3,3) = 19$ and answer two questions posed in [2] by giving examples of minimal super Arrovian domains that are not minimum, and by constructing distinct minimum super Arrovian domains for the same m and n that are not related by a permutation of voters and alternatives.

2. Lower bounds on $\sigma(m,n)$. A *doubles profile* is an n-tuple of linear orders on a two element subset of X. Given a profile $d \in \mathbf{S}^n$ and a two-element subset of the alternatives $Y \subset X$, the restriction $d|_Y$ is a doubles profile. This doubles profile represents the preferences of the voters on the two alternatives of Y in the profile d. A doubles profile is said to be *nonunanimous* if the n voters do not all agree on the ordering of the two alternatives. Let P denote the set of all nonunanimous doubles profiles. A domain $D \subseteq \mathbf{S}^n$ satisfies the *near-free doubles condition* if every nonunanimous doubles profile appears as the restriction of some profile $d \in D$. Fishburn and Kelly found the following necessary and sufficient conditions for a domain to be super Arrovian using the near-free doubles condition.

THEOREM 3 ([2],Theorem 1). *A domain D is super Arrovian if and only if it is Arrovian and satisfies the near-free doubles condition.* □

We begin with a lemma generalizing the "only if" part of this result, motivated by [2, Proof of Part 2 of Lemma 2]. If $S \in \mathbf{S}$ and xSy, then we say that x and y are *nonadjacent* if there is a $w \in X$ such that xSw and wSy. Otherwise, x and y are said to be *adjacent.*

Lemma 4. *Given a super Arrovian domain D, a nonunanimous doubles profile p, and a voter v, there is a profile $d \in D$ restricting to p such that the two alternatives of p are nonadjacent in d for voter v.*

Proof. For concreteness, let v be voter 1, let p be a doubles profile on $\{x, y\} \subset X$, and suppose that x is preferred to y in p for voter 1. Let $C \subset D$ be the set of profiles that restrict to p, and assume that each profile of C has x and y adjacent for voter 1. Then we can define a (P) + (IIA) function $f : D \to \mathbf{R}$ by letting voter 1 be a dictator on $D - C$ and letting voter 1 be a dictator on C *except* for reversing the order of x and y. In other words, f chooses the preferences of voter 1 on every doubles profile other than p. Because D is super Arrovian, f must have a dictator. Since C is nonempty by Theorem 3, voter 1 is not a dictator. Without loss of generality, let voter 2 be a dictator. Then voters 1 and 2 agree on all elements of $D - C$, so that no doubles profile of the form $(yx, xy, \dots)$ appears as the restriction of a profile in D. This contradicts Theorem 3 and proves the lemma. □

Corollary 5. *For $m \geq 3$ and $n \geq 2$, we have $\sigma(m, n) \geq \frac{m}{m-2}(2^n - 2)$.*

Proof. Let D be super Arrovian, and let H be the set of all ordered pairs $(p, d) \in P \times D$ such that d restricts to p, and the two alternatives of p are nonadjacent in d for voter 1. By the lemma, $|H| \geq |P| = \binom{m}{2}(2^n - 2)$. Yet by counting the greatest possible contribution of each $d \in D$, we see that $|H| \leq \binom{m-1}{2}|D|$. Combining these inequalities gives the desired result. □

Lemma 6. *For $m, n \geq 3$, we have $\sigma(m, n) > \frac{m}{m-2}(2^n - 2)$.*

Proof. We need only eliminate the possibility that equality holds in the corollary above. Assume that there exists a super Arrovian domain D of this size. The proof of the corollary implies that for any profile $d \in D$, the alternatives in every Pareto relationship must be adjacent for every voter (otherwise, the contribution of d to $|H|$ is less than $\binom{m-1}{2}$). We claim that this restriction makes it possible to define a (P) + (IIA) function $f : D \to \mathbf{S} \subset \mathbf{R}$ by choosing, for every profile, the reverse of voter 1's preferences on every pair of alternatives except those involved in a Pareto relationship. That is, for $d = (S_1, \dots, S_n)$ we let $f(d)$ be the linear order $\succ_d$ defined by

$$x \succ_d y \text{ if } (x \gg y \text{ in } d) \text{ or } (yS_1x \text{ but not } y \gg x \text{ in } d).$$

Suppose that there is some profile $d = (S_1, \dots, S_n) \in D$ for which this definition of $f(d)$ results in a relation that is not transitive. Then there are three alternatives $x, y, z \in X$ such that the rule defining f yields $x \succ_d y \succ_d z \succ_d x$. First suppose that none of xy, yz, or zx is a Pareto pair. Then we must have yS_1x, zS_1y, and xS_1z, contradicting the transitivity of S_1. Hence we may assume without loss of generality that zx is a Pareto pair and that the others are not. But then we have zS_1y and yS_1x, and the alternatives of the Pareto relationship are nonadjacent. Since this contradicts the assumption about D, we conclude that $f(d)$ is indeed transitive for each profile $d \in D$.

The map f is clearly a (P) + (IIA) function, and voter 1 is not a dictator for f. If there is some other dictator i, then we can choose any other voter j and note that D does not contain any doubles profiles for which 1 and i agree but j differs. This contradicts the near-free doubles condition for D and concludes the proof of the lemma. □

3. An upper bound on $\boldsymbol{\sigma(m, n)}$. We begin by generalizing some notation used by Fishburn and Kelly [2]. Given a subset A of the set $\{1, \dots, n\}$ of all voters and distinct $x, y \in X$, we write xyA for the doubles profile in which the preference ranking of each voter in A is xy and the preference ranking for each voter not in A is yx. Write $\overline{A}$ for $\{1, \ldots, n\} - A$, and note that $xyA = yx\overline{A}$. For a (P) + (IIA) function f on a domain D, we write xAy if $x \succ_d y$ for each $d \in D$ restricting to xyA. If xAy for all ordered pairs of distinct alternatives $x, y \in X$, we write that $A+$ holds. For example, the f constructed in the proof of Lemma 6 has $A+$ if and only if $1 \notin A$. A domain D is super Arrovian if and only if the only (P) + (IIA) functions f on any domain containing D are those defined by: "$A+$ holds whenever $i \in A$" for some fixed i.

LEMMA 7. *For $m \geq 3$ and $n \geq 4$, we have*

$$\sigma(m, n) \leq \sigma(m, 2)(2^{n-1} - 1) + 2^n + \frac{n^2 - 5n - 4}{2}.$$

Proof. We shall construct a super Arrovian domain D of the required size.

Let $\{A, \overline{A}\}$ be a partition of $\{1, \ldots, n\}$, where neither A nor $\overline{A}$ is empty. If $d' = (S_1', S_2')$ is a profile on m alternatives and 2 voters, let $g(d')$ be the profile on m alternatives and n voters $(S_1, \ldots, S_n)$ where $S_i = S_1'$ if $i \in A$ and $S_i = S_2'$ if $i \in \overline{A}$. Consider the $\sigma(m, 2)$ profiles of the form $g(d')$ as d' ranges through a minimum super Arrovian domain on m alternatives and 2 voters. Any (P) + (IIA) function f on a domain containing these profiles must have either $A+$ or $\overline{A}+$. Let Z_1 be the set of all such profiles, for the $2^{n-1} - 1$ possibilities for $\{A, \overline{A}\}$. Note that Z_1 satisfies the near-free doubles condition.

Choose three distinct alternatives $x, y, z \in X$. For each pair $i, j \in \{1, \ldots, n\}$ with $i < j$, consider any profile d in which voter i has preference ranking zxy, voter j has preference ranking yzx, and all the other voters have preference ranking zyx. Let f be a (P) + (IIA) function on a domain containing this profile. By the Pareto condition, $z \succ_d x$. Now $\{i\}+$ would imply $x \succ_d y$, and $\{j\}+$ would imply $y \succ_d z$. Since these three orderings are incompatible, this profile shows that $\{i\}+$ and $\{j\}+$ cannot both hold for f. Let Z_2 be any domain containing such a profile d for every pair i, j. It is clear that we can choose Z_2 such that $|Z_2| \leq \binom{n}{2}$.

Finally, for each subset $A \subset \{1, \ldots, n\}$ with $2 \leq |A| \leq n - 2$, choose an element $i \in A$, and write $B = A - \{i\}$. Consider any profile d in which voter i has preference ranking xzy, the voters in B have preference ranking yxz, and all the other voters have preference ranking zyx. If f is any (P) + (IIA) function on a domain containing this profile, then $\overline{\{i\}}+, \overline{B}+$, and $A+$ cannot all hold for f, since otherwise $y \succ_d x, z \succ_d y$, and $x \succ_d z$. Let Z_3 be any domain containing such a d for each set A. We can choose Z_3 so that $|Z_3| \leq 2^n - (2n + 2)$.

Now let $D = Z_1 \cup Z_2 \cup Z_3$. The domain D has the required number of profiles and satisfies the near-free doubles condition; it remains to show that it is Arrovian. We will refer to the profiles of each Z_i as the "stage i profiles." Let f be a (P) + (IIA) function on D. By the stage one profiles, we have either $\{i\}+$ or $\overline{\{i\}}+$ for each i. The stage two profiles show that for no two distinct i and j can we have both $\{i\}+$ and $\{j\}+$.

Suppose first that $\overline{\{i\}}+$ holds for all i. We show by induction that $\overline{A}+$ holds for all $A \subset \{1, \ldots, n\}$ with $1 \leq |A| \leq n - 2$. For any A with $2 \leq |A| \leq n - 2$, choose the $i \in A$ and $B \subset A$ from the third stage of the construction. Note that $\overline{\{i\}}+$ and $\overline{B}+$ hold by induction, so that $A+$ is impossible by the stage three profiles. Then by the stage one profiles, $\overline{A}+$ must hold, completing the induction. However, this is a contradiction for any $|A|$ with $n/2 \leq |A| \leq n-2$ since $A+$ and $\overline{A}+$ are not compatible.

Hence we must have $\{i\}+$ for exactly one voter i. We must show that this voter is a dictator, that is, $A+$ holds for all subsets A containing i. The assumption about voter i implies that this is true for $|A| = 1$ or $|A| = n - 1$. The inductive argument in the preceding paragraph shows that $\overline{A}+$ holds for each subset A not containing i with $2 \leq |A| \leq n - 2$. This is the desired result. □

Note that the proof of Lemma 7 above works for $n = 3$ except for the paragraph that precludes the possibility that $\overline{\{i\}}+$ holds for all i, since now $n/2 > n-2$. Here one extra profile is needed, namely any profile restricting to (xzy, yxz, zyx), to complete the argument. Henceforth, we will refer to this as a stage three profile in the case $n = 3$. We obtain $\sigma(m, 3) \leq 3\sigma(m, 2) + 4$, which is the result of [2, Lemma 7].

The profiles of stages two and three are wasteful in the sense that we only draw information from three of the alternatives in each profile. For example, given a stage two profile, we can choose some other pair i', j' and alternative w and alter the position of w so that the profile restricts to zxw for voter i', wzx for voter j', and zwx for the other voters. This saves one profile from Z_2. Continuing in this way, we can choose Z_2 such that $|Z_2| \leq \lceil \binom{n}{2}/(m-2) \rceil$. Similarly, for any stage three profile, we can choose two unused alternatives u and v, another triple i', B', A', and alter the profile so that it restricts to uzv for voter i', vuz for the voters in B', and zvu for the other voters. Using up all the alternatives efficiently in this manner allows us to choose Z_3 such that $|Z_3| \leq \lceil (2^n - 2n - 2)/\lfloor m/2 \rfloor \rceil$. We have proven:

Corollary 8. *For $m, n \geq 3$,*

$$\sigma(m, n) \leq \sigma(m, 2)(2^{n-1} - 1) + \left\lceil \frac{n(n-1)}{2(m-2)} \right\rceil + \left\lceil \frac{2^n - 2n - 2}{\lfloor m/2 \rfloor} \right\rceil + 1.$$

For $n \geq 4$, the bound can be decreased by 1. □

4. An improved bound for three alternatives. We now present a separate but similar construction of super Arrovian domains which will allow us to find a stronger upper bound in the case $m = 3$. A domain D is a said to be *basic* if any (P) + (IIA) function f on any domain containing D satisfies $\{i\}+$ or $\overline{\{i\}}+$ for every i, and such that $\{i\}+$ and $\{j\}+$ cannot both hold for distinct i and j.

A domain D is said to be *recursive* for a subset $A \subset \{1, \ldots, n\}$ with $1 \leq |A| \leq n-2$ if for every ordered pair of alternatives $x, z \in X$, there exists an alternative $y \in X$ and a profile $d \in D$ such that d restricts to xyz for the voters in A, yzx for some non-empty proper subset of the other voters, and zxy for the rest. If D is recursive for every A with $1 \leq |A| \leq n - 2$, we say that D is recursive. A domain D which is both basic and recursive is called *inductive*. Note that any inductive domain satisfies the near-free doubles condition.

Lemma 9. *Any inductive domain D is super Arrovian.*

Proof. Let D be inductive. Since D is basic, any (P) + (IIA) function f on D must have $\{i\}+$ or $\overline{\{i\}}+$ for each i. Also, there can be no two distinct i and j such that both $\{i\}+$ and $\{j\}+$ hold. First suppose that $\overline{\{i\}}+$ holds for all i. We shall show by induction on $|A|$ that $\overline{A}+$ holds for each A with $1 \leq |A| \leq n - 1$. The base case, when $|A| = 1$, follows from our assumption. For larger $|A|$, let $x, z \in X$; we need to show that $x\overline{A}z$ holds. Consider the profile $d \in D$ which restricts to xyz for the voters in $\overline{A}$, yzx for the voters in B, and zxy for the voters in C, where $B \cup C = A$ and B and C are nonempty. By the induction, we have $x\overline{B}y$ and $y\overline{C}z$, from which it follows that $x \succ_d y \succ_d z$. Hence we conclude that $x\overline{A}z$ holds, and the induction is complete. Now, the assertions $A+$ and $\overline{A}+$ are contradictory for any A, so we must have $\{i\}+$

for exactly one i. Repeating the induction above shows that $\overline{A}+$ holds for each A not containing i, so that i is a dictator. Hence D is super Arrovian. ∎

THEOREM 10. *For $n \geq 2$, we have $\sigma(3,n) \leq 3(2^n - 2) + 6\binom{n}{\lfloor (n+1)/2 \rfloor} + \binom{n}{2}$.*

Proof. We construct an inductive domain D of the required size. We begin by letting Y_1 consist of the stage two profiles from the proof of Lemma 7 and the $n \cdot \sigma(3,2)$ stage one profiles corresponding to $\{A, \overline{A}\} = \{\{i\}, \overline{\{i\}}\}$, for $i = 1, \ldots, n$. Y_1 is basic, with $|Y_1| = 6n + \binom{n}{2}$.

For any subset $B \subset \{1, \ldots, n\}$ with $2 \leq |B| \leq n-1$ and any $i \in B$, consider the 6 profiles which have the following form, as a, b, and c range over the permutations of the alternatives x, y, and z: the voters in $B - \{i\}$ have preference ranking abc, voter i has preference ranking bca, and the rest of the voters have preference ranking cab. Any domain containing these profiles will be recursive for $B - \{i\}$ and $\overline{B}$. Hence if there is a way to choose, for each subset B with $2 \leq |B| \leq \lfloor \frac{n+1}{2} \rfloor$, an element $i \in B$ such that the sets $B - \{i\}$ range over all possible subsets of size between 1 and $\lfloor \frac{n-1}{2} \rfloor$, then we can create a recursive domain Y_2 containing the corresponding

$$6 \sum_{i=2}^{\lfloor (n+1)/2 \rfloor} \binom{n}{i} = \begin{cases} 3(2^n + \binom{n}{n/2} - 2n - 2) & \text{if } n \text{ is even} \\ 3(2^n + 2\binom{n}{(n+1)/2} - 2n - 2) & \text{if } n \text{ is odd} \end{cases}$$

profiles. Then $D = Y_1 \cup Y_2$ will be inductive of the required size. We now demonstrate the existence of such a choice.

Let $j < n/2$. Consider the bipartite graph whose vertices consist of the subsets of $\{1, \ldots, n\}$ of size j (partite class X) and $j+1$ (partite class Y), where $A \in X$ is connected to $B \in Y$ if $A \subset B$. We want to show the existence of a matching of X into Y, and we will do so using the Hall Matching Condition. Given any subset of the first partite class $Z \subset X$, write $N(Z)$ for the set of neighbors of vertices in Z. We need to show that $|N(Z)| \geq |Z|$ for every $Z \subset X$. Note that each vertex in Z is connected to $n - j$ vertices of Y. This gives $(n-j)|Z|$ edges between vertices in Z and vertices in Y. Since each vertex in Y has degree exactly $j+1$, we have $|N(Z)| \geq (n-j)|Z|/(j+1) \geq |Z|$. Hence the Hall Matching Condition is satisfied, and a matching exists. ∎

COROLLARY 11. *We have $\lim_{n\to\infty} \frac{\sigma(3,n)}{2^n} = 3$.*

5. Two voters. In this section we show that the lower bound of Corollary 5 is an equality for $n = 2$. Fishburn and Kelly [2, Lemma 2] do this for $(m,n) = (3,2)$. For notational purposes, we will use $X = \{1, \ldots, m\}$, and we will label the two voters I and J, with the preferences of voter I written first in each profile. We simply write xIy for $x\{I\}y$. Also, xIY for $Y \subset X$ means xIy for all $y \in Y$. Finally, as an abuse of notation, we write xIX for $xI(X - \{x\})$. We begin by recalling the construction for $m = 3$ in [2].

LEMMA 12 ([2], Lemma 2). *We have $\sigma(3,2) = 6$.*

Proof. Let D_3 be composed of the six profiles in the table below.

	Profile	Pareto	Conclusion
p_1	$(321, 213)$	$2 \gg 1$	$3I1$ or $2J3$
p_2	$(231, 123)$	$2 \gg 3$	$2I1$ or $1J3$
p_3	$(213, 132)$	$1 \gg 3$	$2I3$ or $1J2$
p_4	$(123, 312)$	$1 \gg 2$	$1I3$ or $3J2$
p_5	$(132, 321)$	$3 \gg 2$	$1I2$ or $3J1$
p_6	$(312, 231)$	$3 \gg 1$	$3I2$ or $2J1$

Note that the near-free doubles condition is satisfied. Let f be a (P) + (IIA) function on D_3. The condition $2 \gg 1$ in the first profile implies $3I1$ or $2J3$. Suppose that the first of these holds for f. The condition $2 \gg 3$ in the second profile implies that $2I1$ or $1J3$. But $3I1$ contradicts $1J3$, so $2I1$ must hold for f. Continuing in this way for each profile, we find that xIy for all distinct $x, y \in X$, so that voter I is a dictator on f. Had we assumed that $2J3$ held in the first profile, we would have similarly found that voter J was a dictator. Hence D_3 is super Arrovian. □

LEMMA 13. *We have* $\sigma(4,2) = \sigma(5,2) = 4$.

Proof. For $m = 4$, let D_4 be the domain below.

	Profile	Pareto	Conclusion
p_1	$(1234, 3412)$	$3 \gg 4$	$1I4$ or $3J1$
p_2	$(2413, 1324)$	$2 \gg 4$	$2I1$ or $1J4$
p_3	$(3142, 4231)$	$4 \gg 2$	$1I2$ or $4J1$
p_4	$(4321, 2143)$	$4 \gg 3$	$4I1$ or $1J3$

Note that the near-free doubles condition is satisfied.

Restricting to the three alternatives $\{1, 2, 3\}$, we obtain four of the six profiles of the minimal super Arrovian domain D_3 above. The only information used to show that D_3 is super Arrovian supplied by the two missing profiles is that if f is any (P) + (IIA) function, then $2I1$ or $1J3$ holds and $1I2$ or $3J1$ holds for f. Yet we can conclude this from our domain D_4 using alternative 4. For example, the Pareto relationship $3 \gg 4$ in p_1 implies that $1I4$ or $3J1$ holds. Similarly, from the other profiles we can conclude that $2I1$ or $1J4$ holds, that $1I2$ or $4J1$ holds, and that $4I1$ or $1J3$ holds. These four conclusions combine to supply the "missing" information above. Hence we can conclude that any (P) + (IIA) function f is dictatorial on the three alternatives $\{1, 2, 3\}$.

Suppose that voter I is dictatorial on $\{1, 2, 3\}$. We can conclude from $4 \gg 3$ in p_4 that $4I2$ and $4I1$ hold. Then in p_2 we have $4 \succ_{p_2} 1 \succ_{p_2} 3$ so that $4I3$ holds. Similarly, $3 \gg 4$ in p_1 implies that $1I4$ and $2I4$ hold. Then from p_3, we have $3I4$. Therefore, voter I is a dictator on all four alternatives. Now suppose that voter J is dictatorial on $\{1, 2, 3\}$. We can conclude from $4 \gg 2$ in p_3 that $4J1$ and $4J3$ hold. Hence from p_1, we see $4J2$ holds. Similarly, $2 \gg 4$ in p_2 implies that $1J4$ and $3J4$ hold. Then from p_4, we have $2J4$. Therefore, voter J is a dictator on all four alternatives. Since either voter I or J is a dictator and the near-free doubles condition is satisfied, D_4 is super Arrovian.

For $m = 5$, let D_5 be the domain below.

q_1	$(15234, 35412)$
q_2	$(24135, 51324)$
q_3	$(53142, 42315)$
q_4	$(43251, 21453)$

Note that the restriction of D_5 to $\{1, 2, 3, 4\}$ is the super Arrovian domain D_4 above. Hence there is a dictator on these four alternatives. Suppose that this dictator is voter I. From $5 \gg 2$ in q_1, we can conclude $5I3$. Then from q_3, we have $5I\{1, 2, 4\}$. Similarly, the condition $2 \gg 5$ in q_4 shows that $3I5$, which implies that $\{1, 2, 4\}I5$ from q_2. Hence voter I is dictatorial on all five alternatives. Now suppose that voter J is dictatorial on $\{1, 2, 3, 4\}$. From $5 \gg 4$ in q_1, we can conclude $5J1$. Then from q_2, we have $5J\{2, 3, 4\}$. Similarly, the condition $4 \gg 5$ in q_4 shows that $1J5$, which implies that $\{2, 3, 4\}J5$ from q_3. Hence voter J is dictatorial on all five alternatives. Therefore, D_5 is super Arrovian. □

Note that since $\sigma(m,n) = \frac{m}{m-2}(2^n - 2)$ for $(m,n) = (4,2)$, the proof of Corollary 5 uniquely determines the preferences of each voter, up to permutations of the alternatives, in a minimal super Arrovian domain. The same is true for $m = 6$, and one finds that the super Arrovian domains for $m = 4$ and $m = 6$ are unique up to permutations of the alternatives.

LEMMA 14. *We have* $\sigma(6,2) = \sigma(7,2) = 3$.

Proof. For $m = 6$, let D_6 be the profile below.

p_1	$(123456, 563412)$	(id, τ)
p_2	$(536142, 426153)$	$(\pi, \pi\tau)$
p_3	$(462513, 132546)$	$(\pi^2, \pi^2\tau)$

If we define permutations $\pi = (154)(236)$ and $\tau = (15)(26)$, then the profiles have the form given in the table above, with $\pi^3 = \tau^2 = \mathrm{id}$. The permutation τ switches the voters, and the permutation π rotates the profiles. This symmetry will be important in proving that D_6 is Arrovian.

Let f be a (P) + (IIA) map on D_6. The profile p_1 has the three Pareto pairs $1 \gg 2, 3 \gg 4$ and $5 \gg 6$. Hence, the first alternative in $\succsim_{p_1}$ is 1, 3, or 5, or some combination of these in a tie. Suppose that 3 is in the first position, with or without a tie. Then $3I6$ holds, implying that the order $\succ_{p_2}$ restricts to 5361 on those alternatives. Hence $5I1$ holds, so that the order $\succ_{p_3}$ restricts to 2513 on those alternatives. Thus $2I3$ holds, contradicting the fact that 3 appears first in $\succsim_{p_1}$.

Now suppose that 1 appears first in $\succsim_{p_1}$, possibly tied with 5. Then $1I4$ holds. By the symmetry of the profiles, 5 or 4 appears first in $\succsim_{p_2}$, but 4 cannot appear first because we have $1I4$. Hence 5 appears alone in first in $\succsim_{p_2}$, so we conclude that $5I\{1,2,4,6\}$. Then we have $5 \succ_{p_3} 1 \succ_{p_3} 3$, so $5I3$ as well, and hence $5IX$.

In summary, by assuming that 1 appears first in $\succsim_{p_1}$, we are able to show that 5 appears alone in first in $\succsim_{p_2}$ and that $5IX$ holds. Then by symmetry, we also have that 4 appears alone in first in $\succsim_{p_3}$ and that $4IX$ holds, and that 1 appears alone in first in $\succsim_{p_1}$ and that $1IX$ holds. Now $4I5$ implies $3I\{5,6\}$ from p_1, so that $3IX$ holds from p_2. Similarly, we conclude that $2IX$ and $6IX$ hold, and voter I is a dictator for f.

It follows by symmetry that if we had assumed that 5 appeared first in p_1, then we would have found that voter J was a dictator. Hence D is Arrovian, and since it satisfies the near-free doubles condition, it is super Arrovian.

This domain can be extended to a super Arrovian domain D_7 for $m = 7$ by inserting alternative 7 as the middle preference of each voter in all three profiles (e.g. between the 3 and 4 for both voters in p_1). The domain D_7 is clearly Arrovian since there is a dictator on $\{1, \ldots, 6\}$ and the place of alternative 7 is uniquely determined by the Pareto relationships with the alternatives directly before and after it. Since the near-free doubles condition is satisfied, D_7 is super Arrovian. ∎

This method of inserting alternatives into super Arrovian domains is the fundamental technique used in showing $\sigma(m,2) = 3$ for $m \geq 8$. We state without proof a few more base cases whose verifications are nearly identical to the "insertion" arguments used already.

LEMMA 15. *The domain* D *below, and its restrictions to* $\{1, \ldots, 7\} \cup K$ *for* $K \subseteq \{8, 9, \alpha, \beta, \delta, \gamma\}$*, are all super Arrovian:*

$$\begin{aligned} D = \{&(\beta 1 \delta 2387 \alpha 45 \gamma 69,\ 95 \alpha 63 \gamma 7 \delta 4182 \beta), \\ &(\alpha 5936 \delta 7 \gamma 14 \beta 28,\ 84 \gamma 26 \beta 7915 \delta 3 \alpha), \\ &(\gamma 486297 \beta 51 \alpha 3 \delta,\ \delta 1 \beta 32 \alpha 785496 \gamma)\}. \end{aligned}$$ ∎

A super Arrovian domain D of size 3 for $n = 2$ is called *extremely expandable* if an alternative can be inserted to yield a super Arrovian domain such that for some profile, the new alternative is in first place for voter I and in last place for voter J. Let T be the set of m for which there exists an extremely expandable D for m alternatives. The lemma above implies that $\{7, 8, 9, 10, 11, 12\} \subseteq T$. If $U = \{m : \sigma(m, 2) = 3\}$, then $T \cup (T + 1) \subseteq U$.

LEMMA 16. *If* $m \in T$*, then* $\{m + 8, m + 9, m + 10\} \subset T$.

Proof. Let $D = \{(l_1, l_2), (l_3, l_4), (l_5, l_6)\}$ be super Arrovian for m alternatives, where an alternative A can be inserted to yield the super Arrovian domain $D' = \{(n_1, n_2), (n_3, n_4), (n_5, n_6)\}$ with $n_5 = Al_5$ and $n_6 = l_6A$. Then consider the domain with $m + 8$ alternatives:

$$\begin{aligned} D'' = \{&(1n_1237456,\ 5637n_2412),\\ &(536n_37142,\ 426715n_43),\\ &(462751A3l_5,\ l_61A327546)\}. \end{aligned}$$

We first show that D'' is Arrovian. Let f be a (P) + (IIA) function on D'', and let B be any alternative of D. The previous lemma shows that there is a dictator for f on $\{1, \ldots, 7, B\}$, say voter I. Restricting to the alternatives of D', there is also a dictator since D' is super Arrovian. But in the third profile, $A \gg 3$ and $3 \succ B$, so $A \succ B$, and this second dictator must also be voter I. Also, the conditions $1 \gg A \gg 3$ in the third profile uniquely determine the position of A for this profile. Since this argument holds for every B, we see that voter I is in fact a dictator on all the alternatives. Similarly, if we had assumed that voter J was a dictator on $\{1, \ldots, 7, B\}$, then we would have found that voter J was a dictator on all the alternatives.

Except for possibly the doubles $(3A, A3)$ and $(1A, A1)$, the near-free doubles condition holds for D'' because it holds on D' and the restriction of D'' to $\{1, \ldots, 7, B\}$ for every B. Since these two doubles do appear in the second profile, D'' is super-Arrovian. D'' is extremely expandable because an alternative can be inserted in the position of γ in the previous lemma to yield a super Arrovian domain. Hence $m + 8 \in T$. To show that $m + 9, m + 10 \in T$, one inserts alternatives 8 and 9 into D'' as indicated by the previous lemma. □

COROLLARY 17. *For* $m \geq 8$*, we have* $\sigma(m, 2) = 3$.

Proof. The previous lemma and the base cases $\{7, \ldots, 12\} \subset T$ imply that $\{m \geq 7 : m \neq 13, 14\} \subseteq T$. Hence $\{m \geq 6 : m \neq 14\} \subseteq U$. For $m = 14$, the standard insertion argument shows that

$$\begin{aligned} D = \{&(\beta\omega1\delta2387\alpha45\gamma69,\ 95\alpha63\gamma7\delta418\omega2\beta),\\ &(\alpha5936\delta7\omega\gamma14\beta28,\ 84\delta26\beta791\omega5\delta3\alpha),\\ &(\gamma486297\beta51\alpha3\delta\omega,\ \omega\delta1\beta32\alpha785496\gamma)\} \end{aligned}$$

is super Arrovian, so $U = \{m \geq 6\}$. □

Corollary 8, Lemmas 6, 12, 13, and 14, and Corollary 17 combine to prove Theorems 1 and 2.

6. Miscellaneous results and conclusion. In this section we answer some questions asked by Fishburn and Kelly [2].

PROPOSITION 18. $\sigma(3, 3) = 19$.

Proof. Theorem 2 shows that $\sigma(3, 3) \geq 19$. We create a super Arrovian set of size 19 by starting with the construction of Corollary 8 and eliminating three unnecessary stage one profiles. Let

$$\begin{aligned} D = \{&(zyx, yxz, yxz), & &(yxz, zyx, yxz), & &(yxz, yxz, zyx), & &(yzx, zyx, zxy),\\ &(yzx, xyz, xyz), & &(xzy, yxz, xzy), & &(xyz, xyz, yzx), & &(yxz, yzx, xyz),\\ &(yxz, xzy, xzy), & &(zxy, xzy, zyx), & &(xzy, xzy, yxz), & &(xyz, zxy, xzy),\\ &(xzy, zyx, zyx), & &(zyx, xzy, zyx), & &(zxy, zxy, xyz), & &(xyz, yzx, zxy),\\ &(zxy, yzx, yzx), & &(yzx, zxy, yzx), & &(zyx, zyx, xzy)\}. \end{aligned}$$

The first three columns are stage one profiles for the three partitions $\{\{1\},\{2,3\}\}$, $\{\{2\},\{1,3\}\}$, and $\{\{3\},\{1,2\}\}$, respectively. There are three stage one profiles missing from D: (xyz, zxy, zxy), (xyz, yzx, xyz) and (zyx, zyx, xzy). The fourth column contains the profiles from stages two and three, with some permutation of the alternatives applied for each profile. Note that permuting the alternatives in the profiles for stages two and three does not affect the proof of Corollary 8.

The information provided by (xyz, zxy, zxy) is that any (P)+(IIA) function on a domain containing this profile has $x1z$ or $z\{2,3\}y$. However, we can conclude this directly from the profiles in D. Suppose that $z\{2,3\}y$ does not hold. Then in the first profile of the fourth column, we have $y \succsim z \succ x$, so that $y\{1,2\}x$ holds. Hence $x3y$ does not hold, and using the argument of Lemma 12 on the profiles in the third column, we find $y\{1,2\}z$. Similarly, the third profile of the fourth column and the profiles of the second column show $x\{1,3\}y$. Then in the last profile of the fourth column, we have $x \succ y \succ z$, so that $x1z$ holds. Therefore, the information provided by (xyz, zxy, zxy) is already provided by the profiles in D, and this profile is unnecessary. A similar argument works for the other two missing profiles. Since the near-free doubles condition is satisfied, D is super Arrovian. □

PROPOSITION 19. *There exists an infinite family of minimal super Arrovian domains that are not minimum.*

Proof. Fix $n = 2$, and let $m \geq 4$. Let x and y be the greatest even and odd integers less than or equal to m, respectively. Define E to be the string of alternatives $567 \cdots m$, let $F = 68 \cdots x$, and let $G = 579 \cdots y$. For $L = E, F$, or G, we let $\overline{L}$ be the string of alternatives of L in reverse order. Finally, define

$$\begin{aligned} D = \{&(\overline{E}1234, 3412E),\\ &(3142E, \overline{E}4231),\\ &(2\overline{F}41G3, 1\overline{G}32F3),\\ &(4\overline{G}32F1, 2\overline{F}14G3)\}. \end{aligned}$$

For $m = 4$, we have $D = D_4$ from Lemma 13, so D is super Arrovian. The standard insertion argument shows that D is super Arrovian for larger m as well.

For $m \geq 6$, removing the first profile causes D not to be super Arrovian because Lemma 4 is not satisfied with $p = (m(m-2), (m-2)m)$ on either voter. Similarly, removing the second profile violates Lemma 4 with $p = ((m-2)m, m(m-2))$. Removing the third profile violates the lemma with $p = (m(m-1), (m-1)m)$ for m even and with $p = ((m-1)m, m(m-1))$ for m odd. The same is true with the cases reversed for the fourth profile. Hence D is minimal. By Theorem 1, D is not minimum for $m \geq 6$, and the proof is complete. □

PROPOSITION 20. *Minimum super Arrovian domains need not be unique up to permutations of the voters and alternatives.*

Proof. Let D' be the restriction of the domain D in Lemma 15 to the alternatives $\{1, \ldots, 9\}$, and let D'' be the restriction of D to the alternatives $\{1, \ldots, 8, \alpha\}$. No matter which permutation of alternatives or voters is applied to D', no profile will have the first choice of each voter in the last place for the other voter. Since D'' does contain such a profile, the two minimum super Arrovian domains are not related by permutations. □

Theorems 1 and 2 along with these propositions essentially answer all of the open questions posed in [2]. The exact value of $\sigma(m, 2)$ is calculated for all m, as well as bounds which show that $\sigma(m, n)$ is $\Theta(1)$ for fixed n and $\Theta(2^n)$ for fixed m. Open questions suggested by Theorem 1 are whether $\sigma(m, n)$ is always decreasing in m for fixed n and whether $\sigma(m, n)$ always reaches $2^n - 1$ for m large enough. Another topic for further study is the behavior of $\sigma(m, n)/2^n$ for fixed m as n grows large. We have also left open the question of precisely which pairs (m, n) have the property that there is a unique minimum super Arrovian set on m alternatives and n voters, up to permutations of the alternatives and voters.

Acknowledgments. This work was done under the supervision of Joseph Gallian at the University of Minnesota, Duluth. The author wishes to thank Joseph Gallian, Daniel Biss, and David Witte for their suggestions and encouragement.

REFERENCES

[1] K. J. Arrow, *Social Choice and Individual Value*, 2nd ed., Wiley, New York, 1963.
[2] P. C. Fishburn and J. S. Kelly, *Super Arrovian domains with strict preferences*, SIAM J. Disc. Math., 11 (1997), pp. 83-95.